

The Model 100 Program Book

By
Terry Kepner and David Huntress



The Model 100
Program Book

by
Terry Kepner and David Huntress



Weber Systems Inc.
Cleveland, Ohio

The authors have exercised due care in the preparation of this book and the programs contained in it. The authors and the publisher make no warranties either express or implied with regard to the information and programs contained in this book. In no event shall the authors or publisher be liable for incidental or consequential damages arising out of the furnishing, performance, or any information and/or programs.

Published by:

Weber Systems, Inc.
8437 Mayfield Road
Cleveland, Ohio 44026

For information or translations and book distributors outside of the United States, please contact WSI at the above address.

The Model 100 Program Book

Copyright© 1984 by Weber Systems, Inc. All rights reserved under International and Pan-American Copyright Conventions. Printed in United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise without the prior written permission of the publisher.

ISBN 0-938862-26-X

TRS-80 Model 100 is a trademark
of Radio Shack a division of Tandy Corporation

Cover photo courtesy of Radio Shack
a division of Tandy Corporation

Illustrations by Don Snowden

Table of Contents

PART I — Financial Programs	1
GROWTH Determine the future value and interest earned	3
DEP1 Find the future value of regular deposits	7
CMPND Calculate the interest rate of an installment loan	9
DEPRE Figures depreciation schedules	11
AMORT Calculates an amortization schedule	15
INSTAL Computes monthly payments/number of payments on a loan	19
NOTES Compute maturity value and discount on notes	21
EFFECT Calculate actual versus stated interest	23
BOND Figures actual cost of bonds	25
STOCK Analyze your stock purchase	27
 PART II — Business Programs	 31
MARGIN Calculate percent of mark-up	33
EXPNSE Keep a record of your business travel expenses	37
BARV A vertical bar chart graphic program	41
BARH A horizontal bar chart graphic program	45

PIE	A pie chart graphic program	47
LINE	A graphing program using lines	51
ZFAC	Determine the health of our business	55
PURCHS, INVOICE, STATEM	Create your own billing forms	61

PART III — Home Programs 69

CALNDR	Create a perpetual calendar on your printer	71
LOANS1	Determine the interest rate on a loan	75
PHONE	Separate and tally business calls on your home phone	77
CHECK	Reconcile your checkbook the easy way	81
BUDGET	Set up and track your household budget	85
MILE	Track mileage for your auto	87
MAIN	An automobile maintenance program	91
MAIL	A computerized order form	95

PART IV — Games..... 97

TICTAC	Tic-tac-toe on the Model 100	99
REVERS	A simple analytical game	103
MASTER	Play Mastermind on the Model 100	107
MIMIC	Test your memory	111
WUMPUS	A simple adventure game	113
INVDRS	A Model 100 takeoff of Invaders games	119

PART V — Graphics	121
DIPSY & ART Use math functions to draw geometrical images	123
DOODLE Use the Model 100 display as a drawing board	127
BANNER Create large banners on your printer	129
BIORHY Chart your biorhythms with this program	135
PART VI — Utilities	139
WORDCT Determine the word count of your files	141
LENCTR Find the number of bytes in your BASIC programs	143
CONVR Convert from one numbering system to another	145
MEMSCN Dig around in computer memory to see what's there	149
PRINT Format your BASIC listing as you like	153
PART VII — Sound	155
MUSKEY Make sounds by pressing keys	157
MORSE Learn Morse Code	159
MUSIC8 Create your own tunes	163
PART VIII — Educational	167
HANG Learn how to spell using Hangman	168
TIME How to tell time with a big and small hand	171
TYPE Learn how to touch-type	175
DRILL Set up drill sessions on any subject	179

Introduction

I believe that the TRS-80 Model 100 is the most significant advance in microcomputers since they were invented. Finally, a first time user can sit down at a computer and within moments be using it like a professional. There's no need to learn how to use a complicated disk operating system; no need for time consuming program loading from tape; no worries about saving and loading files; and no need to be tied down to a desk to work.

As a Model 100 owner, you know the advantages of the unit's portability and its built-in word processor, telecommunications program, BASIC, schedule file and address file. What it doesn't have is a library of useful programs that can either be kept in RAM memory, or stored on tape for future use. Radio Shack released the unit with no additional software programs.

That's what started this book: the need for a library of programs. The longer I owned my Model 100, the more programs I accumulated. When I wanted to do something, I just sat down and wrote a program to do it. Unfortunately, the majority of Model 100 owners aren't accomplished BASIC programmers; most have never owned a computer before. Everytime I talked to a Model 100 owner and described a few of the programs I'd written for myself (a program to tell me how long my BASIC programs were; another to tell me how long my text files were, in characters or words; a third to display ROM and RAM memory, etc.), he wanted one or several of them for his own use. Or he'd ask me to write a specific program for him, since he didn't know BASIC well enough to do it himself.

When I mentioned this phenomenon to my friend, Dave Huntress, he suggested we join forces and put a book together. A little more discussion sealed the deal and we were off and running.

The result is what you are currently reading: a book of fun, functional, fascinating programs, and all tailored expressly for the Model 100.

The book is divided into eight sections: finance, business, home, games (every computer should have a few games for it), graphics, utilities, sound and educational. The financial programs should take care of most of your financial questions. I used the amortization program to find out how much principal I've paid off on my home since I bought it. The business programs will assist most businesses, giving them four different methods of graphing data, as well as a pricing margins program and a program for determining how close, or far, their business is from bankruptcy. The home programs cover your checkbook, phone bill, budget, and other aspects of home finance. The games are entertaining, although the logic required for a good game-playing program takes up a lot of room. The three graphics programs are simple (every computer should have a banner program for creating those room-length computer-paper message banners). Two of the utility programs will probably end up always in memory in your computer; they calculate the length of text files and BASIC programs, features left out of the Model 100. The sound programs are easy to use, and fun to try. Finally, the educational programs will keep the kids happy, although the foreign language drill program can be used by adults seeking to improve their skills.

Most of the programs will fit easily in an 8K computer. In fact, the programs average just about 1K in length, leaving enough room for several in memory simultaneously. Some programs, while they'll load into an 8K computer, require more than the available 5460 bytes to operate. These are clearly marked. In fact, all the programs begin with a REM statement that tells you how much room the program occupies in memory (not how much room they need to run). One of them, BANNER, can be made to run in an 8K machine just by removing the special characters, leaving the alphabet and numbers for your use.

One important point to remember is that these programs are tools, to be used, studied and dissected. For the beginner programmer, a variety of programming methods have been used, from brute force calculations to simple and elegant subroutines. We deliberately tried to use different approaches in similar programs to show you that there's more than one approach to solving a problem, and writing a program.

You'll notice that all the programs start with "MAXFILES=0" and a "CLEAR256." This is done to maximize the amount of memory available to the program. BASIC remembers how many files you opened last time you used them and keeps them in memory until you expressly tell it to forget them. This is why the unit tells you it has only 5062 when you first turn it on. If you go to BASIC, type "CLEAR0:MAXFILES=0," and go back to the menu, you'll see that it now says you have 5460 bytes free. The difference between the two lies in the number of files retained by BASIC, plus miscellaneous pointers.

As you look through the programs you'll notice the liberal use of subroutines,

and those who do know a little programming will notice the lack of “structured” programming. This was done for one reason: room. Structured programming is nice when you have the memory to waste, but when every byte counts, and the lack of ten can prevent the use of a program, you have to write as tightly as possible.

In this book we’ve written many of the programs with commands on separate lines, but in other programs we’ve written the command logic to pack as much as possible in one line. You’ll notice this the most with WORDCT and LENCTR, the two programs for determining the length of text and BASIC files. That’s because we thought those two would be the most likely to be in memory all the time, so we tried to make them as compact as possible. (Everytime you remove a line number by combining two lines into one, you remove four bytes from the length of your program. Reduce the number of lines from 100 to 10 and you’ve saved a considerable amount of space for other programs.

About half the programs were written by Dave on a TRS-80 Model 1. When he was ready, he shipped them over the phone line to my Model 100, where I added the finishing touches. It’s a tribute to the design of the Model 100 that Dave had more problems getting his Model 1 RS232 board to work right than I had with my portable.

Two programs were not written by Dave or me: ZFAC and MUSIC8. “ZFAC” is the bankruptcy analysis program. This program was originally written by Mark Robinson for his Model I. The other program, MUSIC8, was written by Bradford P. Stevens. Both are exceptional programs, and we are indebted to Mark and Brad for letting us use their programs in our book.

Happy computing, folks!

About the Listings

The listings in this book were created using the program PRINT which is also included as one of the programs. PRINT is a print formatting program which allows you to set the page and line length. It also allows you to indent continuation (wrap around) lines if the line length exceeds the number of characters you have specified.

Since the screen of the Model 100 will display only forty characters, we have listed the program lines at that number. This way, when you type in the programs, the screen should wrap around at the same time the printed listing does. This is true for the first line of the listing only.

For clarity in the listings, we have indented all wrap around lines by five spaces. *Do not enter these spaces* when you type in the programs.

For example, here is a sample line from one of the programs:

```
90 INPUT"Amount of each payment (dollars  
    )";P
```

Note that the line wraps around, and that there are five spaces preceding the second line, under the 9, 0, space, I and N. Do not enter these spaces when typing in the programs.

PRINT USING statements will also require special care. The backward slash is obtained on the Model 100 by pressing the GRPH key and the minus sign at the same time. To accurately type in the proper number of spaces between backward slashes, count the number of spaces in the line above or below such a line. The listings were produced on a ten-character per inch printer. Be especially careful when PRINT USING statements wrap around so that you get the proper number of spaces.

The "^" character is obtained by pressing SHIFT 6 on the Model 100. This

symbol occurs a few times in the programs and is the symbol for "raise to a power," as in 2^2 (which means two raised to the second power).

Note that the zero is slashed in all the listings, and the letter O is not.

You will notice that key words of the BASIC language are not separated by spaces in any of the listed programs. This makes the programs slightly more difficult to read, but saves valuable memory space.

Every effort has been made to verify the accuracy of the programs listed. They were produced by first running the program on the Model 100, and after successfully running, were printed out directly from the computer to a line printer. The listings were then photographed directly, so that no errors could be introduced by typesetting or transmission.

By paying some attention to these notes you should be able to enjoy using all the programs in this book.

PART I

Financial Programs

Practically everyone, at one time or another, needs to do financial calculations of one kind or another. Things as simple as: "How much do I have to deposit every week at the bank to have \$500 spending money at Christmas?" or "What's the installment payment on a \$200 TV, with \$25 finance charge, a \$50 down payment, and \$16 sales tax?" Sometimes the calculations are more complex, such as "How much money is owing on my house mortgage after 15 years of payments?" (handy in determining your home's equity). Or even in determining if you can afford to buy that dream house you've always wanted.

This section is devoted to programs for handling many of the financial calculations most people need. You won't need them all immediately, but you'll end up using almost all of them, eventually.

There are ten programs in this section. GROWTH will tell you the future value of a present sum, the amount of money needed to be left in deposit for a future sum, and interest earned per day on a given sum. The last is valuable in determining 90-day deposit earnings.

DEP1 will tell you the future value of a series of regular deposits, and the size of regular deposits for a future target sum.

COMPND calculates the interest rate of an installment loan when given the starting and ending values, and the time length of the loan.

The next program, DEPRE, calculates the salvage value, depreciation rate, amount and schedule for a business.

For the home owner, AMORT calculates an amortization schedule for you, with either display or lineprinter output.

INSTAL is a program for calculating the monthly payment, or number of payments, for an installment loan.

For those interested in promissory notes, NOTES will tell you the maturity

value, and the discount.

EFFECT is a short program that calculates the actual interest, versus the stated interest, of an investment.

BOND is another investor's program that calculates the actual cost of the bond after brokerage fees, and the rate of return.

The last program in this section is STOCK, which tells you your stock purchase cost, the sales cost/profit, the dividend, and the yield of your stock.

If you want, and you have the room available, you could combine the programs you find most useful into a larger program by writing a short menu routine listing the features you want, and treating the programs as subroutines. Don't forget to discard the programs' menus, since your new menu replaces them.

GROWTH

Determine future value and interest earned.

This program calculates the future value of a sum of money left on deposit. This is approached in two ways. The first assumes that you have a sum of money invested. You give the program the interest rate, the number of interest computing periods per year, and the number of periods the sum will be invested. The program then tells you the sum of money you'll have at the end of that time.

The other method assumes that you want to know how much money you have to leave on deposit to have a certain amount of money at a specific time in the future. Like the other method, you specify the interest rate paid, the number of interest computing periods, and the total number of periods the money will be left invested.

Both routines require that the interest rate be typed in as whole numbers (5% as opposed to .05). Interest is calculated on the sum at the end of each period, and the money earned is added to the sum for the next period. This is known as compounding. The sum at the end of the time period is called the compound amount.

The last routine calculates the daily interest earned on a sum for the period of days you give it, telling you the ending balance and per day interest.

Program Listing for GROWTH

```
5 REM 1279 bytes
10 CLS: CLEAR 256: PRINT TAB(12) "Earning Int
   erest"
20 PRINT: MAXFILES=0
30 PRINT "1. Present Value of a Future Su
   m"
```

```

40 PRINT"2. Future Value of Present Sum"
50 PRINT"3. Simple Interest, by days"
60 PRINT:PRINTTAB(13)"Your Choice";
70 INPUTA:IFA<1ORA>3THEN10
80 ONAGOSUB90,220,340:GOTO10
90 CLS:PRINTTAB(5)"PRESENT VALUE OF A FU
    TURE SUM"
100 INPUT"Future sum (dollars)";F
110 INPUT"Annual interest rate (percent)
    ";R
120 INPUT"Number of periods in one year"
    ;M
130 INPUT"Num. of periods to be calculat
    ed";N
140 I=R/M:I=I/100
150 T=1+I:A=T
160 IFN=1THEN180
170 FORX=1TO(N-1):S=A*T:A=S:NEXTX
180 P=F/A
190 PRINTUSING"Is worth, in today's doll
    ars:$$##,###.##";P;
200 PRINT@288,"Press <ENTER> to return";
210 IFINKEY$=CHR$(13)THEN10ELSE210
220 CLS:PRINT"    Present Value of Futur
    e Sum"
230 INPUT"Present sum (dollars)";P
240 INPUT"Annual Interest rate (percent)
    ";R
250 INPUT"Number of periods in a year";M
260 INPUT"Number of periods to maturity"
    ;N
270 I=R/M:I=I/100:T=I+1:A=T
280 IFN=1THEN300
290 FORK=1TO(N-1):S=A*T:A=S:NEXTK
300 F=P*A
310 PRINTUSING"The future value is $$##,
    ###.##";F
320 PRINTUSING"The increase is $$##,###.
    ## or####%";F-P;(F/P)*100-100
330 GOTO200
340 CLS:PRINTTAB(10)"Simple Interest"
350 INPUT"Starting Balance (dollars)";P
360 INPUT"For how long (in days)";D
370 INPUT"What interest rate (percent)";
    R
380 I=R/100/365*D*P
390 I=INT(I*100+.5)/100
400 PRINTUSING"Ending Balance = $$###,##

```


###";I+P
410 PRINTUSING"or \$\$\$### interest per d
ay";I/D
420 GOTO200\



Ω



DEP1

Find the future value of regular deposits.

This program reports the future value of regular deposits to a bank, or the deposits required to reach a future value, given the length of time and periodic interest accumulation.

In this program the interest is calculated on the balance with the same frequency that you add your deposits to the balance. That is, if you make twelve deposits a year, the program will compute the interest as compounded monthly. This is known as an annuity due, in which the payments to the balance are made before the interest is computed and paid, much like an insurance policy or savings account.

This kind of calculation is useful when you want to save a predetermined amount of money for a purchase or expense, such as a vacation trip, college tuition, and so forth.

Program Listing for DEP1

```
10 REM 918 bytes
20 CLS: CLEAR 256: MAXFILES=0
30 PRINT TAB(10) "Deposits And Values
40 PRINT: PRINT "1. Regular Deposits, Fut
   ure Value"
50 PRINT "2. Future Value, Regular Depos
   its"
60 PRINT: PRINT TAB(13) "Your Choice";: INPU
   TA
70 IFA=2 THEN 180 ELSE IFA<>1 THEN 10
80 CLS: PRINT "Future Value of Regular
   Deposits"
```

```

90 INPUT"Deposit amounts";D
100 INPUT"Number of deposits per year";P
110 INPUT"Length of time deposits will b
    e made (year, month)";Y,M
120 Y=Y+M/12
130 INPUT"Annual interest rate (percent)
    ";I
140 I=I/P/100
150 T=(1+I)^(P*Y):T=T-1:T=T/I:FV=D*T
160 PRINTUSING"The future value is $$###
    ##,###.##";FV
170 GOTO280
180 CLS
190 PRINT"Regular Deposits Needed for Fu
    ture Value"
200 INPUT"Desired future value (dollars)
    ";F
210 INPUT"Number of deposits per year";P
220 INPUT"Length of time (years, months)
    ";Y,M
230 Y=Y+M/12
240 INPUT"Annual interest rate (percent)
    ";I
250 I=I/P/100
260 T=(1+I)^(P*Y):T=T-1:T=I/T:R=F*T
270 PRINTUSING"Regular deposits needed =
    $$####,###.##";R
280 PRINT@287,"Press <ENTER> to return";
290 IFINKEY$<>CHR$(13)THEN290ELSE20

```

Ω

COMPND

Calculate the interest rate of an installment loan.

Compound Interest: at stated intervals the interest earned on a savings account, or other investment, is computed and added to the principal. Thus, it is converted, or compounded, into the principal and earns interest itself for the remainder of the interest paying periods. The difference between the sum of money at the end of the transaction and the sum at the beginning is called compound interest.

The compound interest rate is the rate at which the principal is compounded.

This program calculates the compound interest rate of loan or investment. You give it the future value, present value, number of interest calculating periods per year, and the length of time for which the interest rate is to be figured. From these numbers, the program figures the percentage rate at which the money is compounding.

Program Listing for COMPND

```
5 REM 937 bytes
10 CLS: CLEAR 256: MAXFILES=0
20 PRINT TAB(10) "Compound Interest"
30 INPUT "Future Value (dollars)"; FV
40 IFF=2 THEN 100
50 INPUT "Present value (dollars)"; PV
60 IFF=1 THEN 100
70 INPUT "Number of periods per year"; PY
80 IFF=3 THEN 100
90 INPUT "Length of time (YY,MM,DD) "; Y,M
    ,D
100 PD=Y*PY+M*PY/12+D*PY/365
110 I=((FV/PV)^(1/PD)-1)*100*PY
```

```

120 A$="###.##":F=0
130 CLS:PRINTTAB(12)"Compound Interest"
140 PRINTUSING"Future Value (dollars)
    $$##,###.##";FV
150 PRINTUSING"Present value (dollars)
    $$##,###.##";PV
160 PRINTUSING"Number of periods per year
    ###";PY
170 PRINTUSING"Length of time ## yrs, ##
    mths, ## days";Y,M,D
180 PRINTUSING"Annual interest rate is
    ###.##%";I
190 PRINT@286,"Press <ENTER> to continue
    ";
200 IFINKEY$<>CHR$(13)THEN200
210 CLS:PRINT"Do you want to:"
220 PRINT"    1 - Change present value"
230 PRINT"    2 - Change Future value"
240 PRINT"    3 - Change periods/year"
250 PRINT"    4 - Change time range"
260 PRINT"    5 - Re-enter the information"
270 PRINTTAB(13)"Your choice";:INPUTF
280 IFF<1ORF>5THEN210
290 F=INT(F):ONFGOTO50,30,70,90,10

```

Ω

DEPRE

figures depreciation schedules.

Most items purchased lose value as time passes. Some because they wear out, others because they become obsolete. The decrease in value of an item over a period of time is called depreciation.

This program figures the salvage value, depreciation rate, depreciation amount, and depreciation schedules for three types of depreciation: straight-line, declining and double-declining.

The first option of this program takes the original cost of the item, the length of its useful lifespan, and uses the depreciation rate to determine the salvage value.

If you don't know the salvage value of the product, but you do know its age, original price and current value, you can calculate the rate of depreciation using the second option of this program.

The third option displays the amount of depreciation by taking the purchase price, the depreciation rate, and the number of years to depreciate, calculating the depreciation for each year and the remaining value of the item.

Finally, for a schedule of depreciation, tell the program the original price, the salvage value, the number of years to depreciate, and the method of depreciation: straight-line, declining, or double-declining. Straight-line depreciation takes the cost of an item, subtracts the salvage value, and divides this amount by the number of years the item is expected to last. The result is the amount of depreciation that can be subtracted from the asset value, and income, of the business each year. Declining depreciation subtracts a percentage from the item's value each year until the salvage value is reached. This has the advantage of higher reductions in value (and more to take off taxes) in the beginning years of depreciation. Double-declining is slanted even more towards higher depreciation in the first few years.

Which of these methods you use is something you should discuss with your accountant.

Program Listing for DEPRE

```
5 REM 1722 bytes
10 CLS: CLEAR 256: MAXFILES=0
20 PRINT TAB(14) "Depreciation"
30 PRINT: PRINT "1. Salvage Value"
40 PRINT "2. Depreciation rate"
50 PRINT "3. Depreciation Amount"
60 PRINT "4. Depreciation Schedule"
70 PRINT: PRINT TAB(13) "Your Choice";
80 INPUT A: IFA<1 OR A>4 THEN 10
90 CLS: ON AGO SUB 100, 170, 240, 380: GOTO 10
100 PRINT TAB(10) "Salvage Value"
110 PRINT: INPUT "Age of item (years)"; Y
120 INPUT "Depreciation Rate (percent)"; D
130 INPUT "Original Price (dollars)"; O
140 PRINT
150 PRINT USING "The Salvage Value is $$#
      #####,###.##"; O*(1-D/100)^Y
160 GOTO 620
170 PRINT TAB(10) "Depreciation Rate"
180 PRINT: INPUT "Current Value (dollars)";
      ;D
190 INPUT "Original Price (dollars)"; O
200 INPUT "Length of time owned (years)";
      Y
210 PRINT
220 PRINT USING "The Depreciation Rate is
      ####.##%"; (1-(D/O)^(1/Y))*100
230 GOTO 620
240 PRINT TAB(10) "Depreciation Amount"
250 INPUT "Purchase Price (dollars)"; P
260 INPUT "Depreciation Rate (percent)"; D
270 INPUT "Number of years to depreciate"
      ;Y
280 D=D/100: DT=0
290 GOSUB 370
300 FOR X=1 TO Y
310 DP=D*P*(1-D)^(X-1)
320 DT=DT+DP
330 PRINT USING "### $$$$$$,###.## $$
      #####,###.##"; X; DP; P-DT
340 IF X/5=INT(X/5) AND Y<>X THEN GOSUB 650
350 NEXT X
```

```

360 GOTO620
370 CLS:PRINT"Year      Depreciation
      Remainder":RETURN
380 PRINTTAB(10)"Depreciation Schedule"
390 PRINT:PRINT"Depreciation Factor: (St
      raight line ="
400 PRINT" 100%, Declining Balance = 101
      -199%,"
410 INPUT" Double declining = 200%)" ;D
420 INPUT"Original Price (dollars)";O
430 INPUT"Salvage Value (dollars)";S
440 INPUT"Number of years to depreciate"
      ;Y
450 PRINT
460 GOSUB370
470 DV=O-S:R=O
480 IFD=100THEN550
490 D=D-100:FORX=1TOY
500 DP=D/100*R:R=R-DP
510 PRINTUSING"###    $$##### ,###.##    $$
      ##### ,###.##";X;DP;R
520 IFX/5=INT(X/5)ANDY<>XTHENGOSUB640
530 NEXTX
540 GOTO620
550 R=DV
560 DP=DV/Y
570 FORX=1TOY
580 R=R-DP
590 PRINTUSING"###    $$##### ,###.##    $$
      ##### ,###.##";X;DP;R+S
600 IFX/5=INT(X/5)ANDY<>XTHENGOSUB640
610 NEXTX
620 PRINT@287,"Press <ENTER> to Return
      ";
630 IFINKEY$<>CHR$(13)THEN630ELSERETURN
640 PRINT@287,"Press <ENTER> to continue
      ";
650 GOSUB630:CLS:GOTO370

```

Ω

LET'S SEE...
\$150,000 AMORTIZED
OVER THIRTY YEARS...



AMORT

Calculates an amortization schedule.

In business mathematics, the term amortization is used to mean the retirement of a debt by means of periodic payments. It is most frequently used to prepare mortgages for home owners, and for businesses making large land, building, or equipment purchases.

This program will prepare a payment schedule for such a purchase, given the principal (amount loaned or borrowed), the number of periods of the loan (in months), the annual interest rate, and the payment, if known. If the payment amount is unknown, which it usually is, the program will calculate an approximation of the monthly payment.

Once the information is input, you're given the choice of having the information displayed on your Model 100, or having it sent to a lineprinter (80-column). The schedule created tells you the payment number, the amount of the principal remaining, the payment amount, and the amount of the payment that is used to pay the current period's interest on the principal. At the bottom of each page printed or displayed, the loan information on principal, length of time, and interest rate are repeated.

The practical limit of this program is \$9,999,999. Any amount larger than this will cause the display to be distorted as the numbers exceed the room allocated to them on the display. Larger values can be calculated with a printer, but the same problem is reached at about \$90,000,000.

If you need to calculate larger values, add number signs (#) to the string variable B\$ in line 90, and change the tabs in lines 440-480 to accommodate the larger numbers.

Program listing for AMORT

```

5 REM 1352 bytes
10 CLEAR256:MAXFILES=0:DEFDBLI,T:CLS
20 PRINTTAB(12)"Amortization Schedule"
30 INPUT"Principal (dollars)";P
40 INPUT"Num. of periods in loan (months
   )";L
50 INPUT"Annual interest rate (percent)"
   ;R
60 PRINT"Payment,"
70 INPUT"if known (enter if unknown)";M
80 INPUT"Do you want this on your printe
   r";A$
90 IFA$="Y"ORA$="Y"THENF=1:B$="$$####,##
   #.##"ELSEF=0
100 I=R/1200
110 T=1-1/(1+I)^L:K=P
120 IFM<>0THEN140
130 M=P*I/T
140 GOSUB330
150 FORZ=1TOL
160 IFF=1ORC<5THEN210
170 PRINTUSING"Loan yr##.##$#####.## ##
   # months at ##%";(Z-1)/12;K;L;R;
180 PRINT@280,"Press <ENTER> to continue
   , 1 to restart";
190 A$=INKEY$:IFA$="1"THEN10ELSEIFA$<>CH
   R$(13)THEN190
200 C=0:GOSUB330
210 A=P*I
220 A=INT(A*100+.5)/100
230 M=INT(M*100+.5)/100
240 B=M-A:P=P-B
250 B=INT(B*100+.5)/100
260 P=INT(P*100+.5)/100
270 IFFTHEN440
280 PRINTUSING"###$#####.##";Z;P;
290 PRINTUSING"$#####.##";M;B;A;
300 C=C+1:NEXTZ:IFFTHENLPRINTCHR$(12)
310 PRINT@286,"Press <ENTER> to start ov
   er";
320 IFINKEY$<>CHR$(13)THEN320ELSE10
330 IFFTHEN360ELSECLS
340 PRINT"Per.Remainder Payment Principa
   l Int."

```

```

350 RETURN
360 IFZ/48<>INT(Z/48)THEN300
370 LPRINT:LPRINT:LPRINT:LPRINT:LPRINT
380 LPRINT"Amortization Schedule";
390 LPRINTTAB(70)USING"Page ##";INT(Z/48
)+1
400 LPRINTTAB(5)USING" - Loan $$#####,##
    #.## - ### months - Interest Rate #
    ##.##% -";K;L;R
410 LPRINT:LPRINT
420 LPRINTTAB(8)"Payment"TAB(20)"Remaind
    er"TAB(37)"Payment"TAB(51)"Principa
    l"TAB(67)"Interest"
430 IFZ=0THENRETURNELSE300
440 LPRINTTAB(10)USING"###";Z;
450 LPRINTTAB(17)USINGB$;P;
460 LPRINTTAB(32)USINGB$;M;
470 LPRINTTAB(48)USINGB$;B;
480 LPRINTTAB(63)USINGB$;A
490 IFZ<>LANDZ/12<>INT(Z/12)THEN300
500 LPRINTTAB(35)USING"Loan year ##";(Z-
    1)/12
510 LPRINT:GOTO360

```


INSTAL

Computes monthly payments/number of payments on a loan.

Installment buying spreads the cost of a purchase across a span of several payments. This program will tell you either the number of payments you must make, given the purchase cost, finance charge, sales tax (if any) and the size of the monthly payment, or the size of the monthly payment given the cost, finance charge, sales tax (if any), and the number of payments to be made.

In addition to this information, the program also tells you the total deferred price you're paying for the item.

Program Listing for INSTALL

```
10 REM 1220 bytes
20 CLS: CLEAR 256: MAXFILES=0
30 PRINT TAB(10) "Installment Loans"
40 PRINT: PRINT " 1. Monthly Payment"
50 PRINT " 2. Number of Payments"
60 PRINT
70 PRINT TAB(13) "Your Choice";: INPUT A
80 CLS: IFA=2 THEN GOSUB 180
90 IFA=1 THEN GOSUB 110
100 GOTO 20
110 PRINT " Finding Monthly Installment
    Payments"
120 GOSUB 320
130 INPUT "How many payments"; P
140 CLS: GOSUB 380
150 PRINT "Monthly Payment" TAB(28) USING "$
    ###,###.##"; (C-D+F+T*C)/P
160 PRINT "Total deferred price" TAB(28) US
```

```

      ING"$$##,###.##";C+F+T*C
170 GOTO440
180 PRINT"Finding the Number of Payments
    "
190 GOSUB320
200 INPUT"What is the monthly payment";M
210 P=(C-D+F+C*T)/M
220 R=0:IFP=INT(P)THEN260
230 Q=INT(P)*M
240 R=(C-D+F+C*T)-Q
250 P=INT(P)
260 CLS:GOSUB380
270 PRINTUSING"You have   ### payments of
      $$$#,###.##";P;M
280 IFR>0THENPRINT"Final Payment is"TAB(
      30)USING"$$,###.##";R
290 GOSUB440:CLS
300 PRINT"Total deferred price"TAB(28)US
      ING"$$##,###.##";C+F+T*C
310 GOTO440
320 INPUT"Cash price (dollars)";C
330 INPUT"Down payment (dollars)";D
340 INPUT"Finance Charge (dollars)";F
350 INPUT"Sales Tax (percent, zero if no
      ne)";T
360 IFT<>0THENT=ABS(T/100)
370 RETURN
380 PRINT"Cash Price"TAB(28)USING"$##,##.##";C
390 IFT>0THENPRINT"Sales Tax"TAB(28)USIN
      G"$##,###.##";T*C
400 PRINT"Down payment"TAB(28)USING"$##,###.##";D
410 PRINT"Unpaid balance"TAB(28)USING"$##,###.##";C-D+T*C
420 PRINT"Balance + Finance Charge"TAB(2
      8)USING"$##,###.##";C-D+F+T*C
430 RETURN
440 PRINT@287,"Press <ENTER> to continue
    ";
450 IFINKEY$<>CHR$(13)THEN450ELSERETURN

```

Ω

NOTES

Compute maturity value and discount on notes.

When you borrow money, you have two ways to pay it off. The most common method is to return the money in periodic payments. The other method is to pay back the lump sum, plus interest, at a specified date in the future.

The loans themselves are handled in two ways: full notes and discounted notes. Full notes give you the full amount of the loan, which you pay back plus interest. The discounted notes are loans with the interest payment deducted from the money you receive.

This program figures both forms of promissory notes.

Program Listing for NOTES

```
10 REM 623 bytes
20 CLS: CLEAR 256: MAXFILES=0
30 PRINT TAB(10) "Promissory Notes"
40 PRINT: PRINT "  1. Maturity Value"
50 PRINT "  2. Discounting Notes"
60 PRINT
70 PRINT TAB(13) "Your Choice";: INPUT A
80 CLS: IFA=2 THEN 140
90 IFA<>1 THEN 10
100 PRINT "    Maturity Value of Promissor
    y Notes"
110 PRINT: GOSUB 190
120 PRINT USING "Maturity Value = $$####, #
    ##.##"; IT+D
130 GOTO 240
140 CLS: PRINT TAB(10) "Discounted Notes"
150 GOSUB 190
```

```

160 PRINTUSING"Discount =          $$#,##
    #.##";IT
170 PRINTUSING"Cash Received = $$####,##
    #.##";D-IT
180 GOTO240
190 INPUT"Amount of note (dollars)";D
200 INPUT"Period of note (days)";P
210 INPUT"Annual interest rate (percent)
    ";I
220 IT=((I/100)*D)*(P/365)
230 RETURN
240 PRINT@287,"Press <ENTER> to return";
250 IFINKEY$<>CHR$(13)THEN250ELSE20

```



Ω

EFFECT

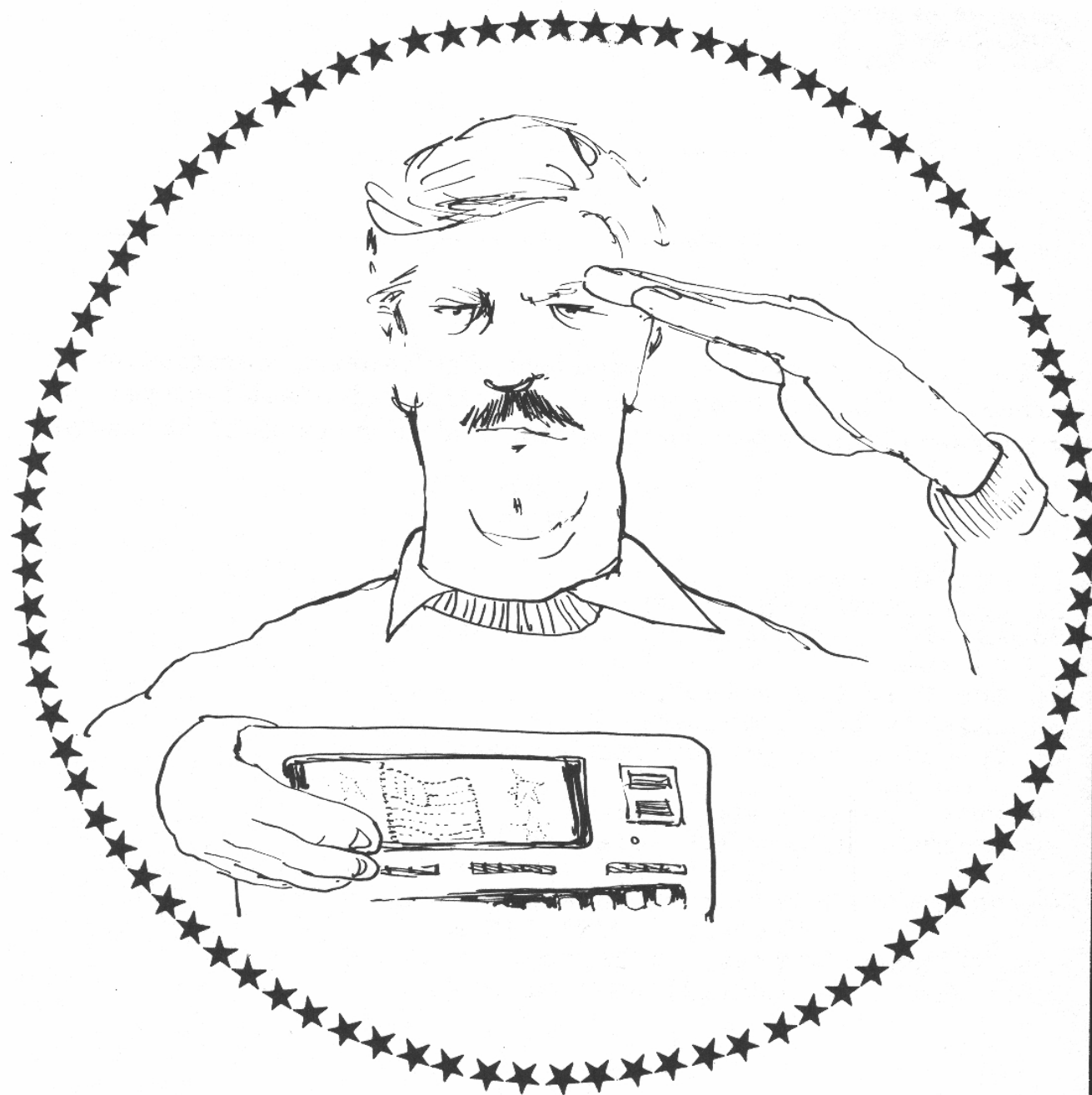
Calculate actual versus stated interest.

This program gives you the nominal and effective interest rates of an investment or loan, based on the number of interest compounding periods per year and the number of periods between the starting and ending values of the amount.

Program Listing for EFFECT

```
10 REM 385 bytes
20 CLS: CLEAR 256: MAXFILES=0
30 PRINT "Nominal & Effective Interest
  Rates"
40 INPUT "Future Value"; F
50 INPUT "Present Value"; V
60 INPUT "Number of periods between the t
  wo"; N
70 INPUT "Periods per year"; P
80 PRINT USING "Nominal Interest Rate is
  ###.##%"; (P * ((F/V)^(1/N)) - P) * 100
90 PRINT USING "Effective Interest rate is
  ###.##%"; ((F/V)^(P/N) - 1) * 100
100 PRINT @287, "Press ENTER to continue";
110 IF INKEY$ <> CHR$(13) THEN 110 ELSE 20
```

Ω



BOND

Figures actual cost of bonds.

If you own or are thinking of buying bonds, some helpful information on your investment is generated by this program. Using the bond cost, number of bonds, interest rate and brokerage fee, it calculates your total cost for the bonds, the interest income you'll receive, and the percentage of return on your investment.

Program Listing for BOND

```
10 REM 622 bytes
20 CLS:MAXFILES=0:CLEAR256
30 PRINTTAB(16)"Bonds"
40 INPUT"Cost per bond (dollars)";C
50 INPUT"Number of bonds";N
60 INPUT"What interest rate (percent)";I
70 INPUT"Bond Par Value";P
80 INPUT"brokerage fee (dollars)";B
90 IFB=0THENINPUT"Brokerage fee (percent)";R
100 F=B:IFB=0ANDR<>0THENF=R/100*N*C
110 CLS
120 PRINTTAB(7)USING"#### bonds at $$###
    .##";N;C
130 PRINT"Bonds cost"TAB(25)USING"$$###,
    ###.##";10*N*C
140 PRINT"Brokerage fee"TAB(27)USING"$$#
    ,###.##";F
150 PRINT"Total Cost"TAB(25)USING"$$###,
    ###.##";N*C*10+F
```

```
160 PRINT USING "Interest Income per year$  
$####,###.##"; P*N*I/100  
170 PRINT TAB(7) USING "Return on investmen  
t is ###.###%"; (P*I/100)/C*10  
180 PRINT@288, "Press ENTER to do again."  
;  
190 IF INKEY$ <> CHR$(13) THEN 190 ELSE 20
```

STOCK

Analyze your stock purchase.

Another source of investment income is stock purchases. This program calculates your stock purchase cost by taking the per unit cost, multiplying by the number purchased, and subtracting the brokerage (either as a percent cost or as a dollar fee).

It also calculates the cost and profit of a stock sale, using the original purchase cost (from option one, for example), the sale price and quantity of stocks sold, the brokerage fee (again either a percentage or flat fee), and the state transfer tax (percentage or flat fee). This information is summarized in a final display for your convenience.

Finally, the program calculates the dividend and yield of your stocks, taking the cost of the stock, its dividends per share, and the number of stocks. The final display gives you your total dividend income, and the rate of return on your investment.

If you're interested, you could probably modify this program to store the stock costs and other information in memory, or tape, and use an input loop to calculate the cost and rate of return on your entire investment portfolio. By assigning variables to the calculated values, and adding to them with each stock read in you could accomplish this without increasing the program beyond your 5K memory constraint. (Don't forget that you can specify CAS:FILE as a file name used in an OPEN statement to store the information on tape instead of just to memory. See the MILE or MAIN programs in the HOME section of this book).

Program Listing for STOCK

```
10 REM 1521 bytes
20 CLS:MAXFILES=0:CLR256
```

```

30 PRINTTAB(10)"Stock Profits/Costs"
40 PRINT:PRINT"1. Stock Purchase Cost"
50 PRINT"2. Stock Sale Cost/Profit"
60 PRINT"3. Dividend and Yield"
70 PRINT:PRINTTAB(13)"Your Choice";:INPU
  TA
80 IFA<1ORA>3THEN20
90 CLS:ONAGOSUB100,160,290:GOTO20
100 PRINTTAB(10)"Stock Purchase Cost"
110 PRINT:INPUT"Cost per share";C
120 GOSUB360
130 GOSUB410
140 PRINT"Total Cost for Stock"TAB(26)US
    ING"$ $###,###.##";N*C+F
150 GOTO490
160 PRINTTAB(10)"Stock Sale"
170 PRINT:INPUT"Original Purchase cost (
    dollars)";PP
180 INPUT"Sale price per stock (dollars)
    ";C
190 GOSUB360
200 INPUT"State Transfer Tax (percent)";
    TT
210 IFTT=0THENINPUT"State Transfer Tax (
    dollars)";ST
220 T=ST:IFTT<>0THENT=TT/100*N*C
230 GOSUB410
240 PRINT"State Tax"TAB(27)USING"$ $###,##
    #.##";T
250 PRINT"Total cash received"TAB(26)USI
    NG"$ $###,###.##";C*N-T-F
260 PRINT"Original purchase cost"TAB(27)
    USING"$ $###,###.##";PP
270 PRINT"Total profit"TAB(26)USING"$ $###
    #,###.##";C*N-T-F-PP
280 GOTO490
290 PRINTTAB(10)"Dividends and Yield"
300 INPUT"Cost of shares (dollars)";C
310 INPUT"Number of shares";N
320 INPUT"Dividends per share (dollars)"
    ;D
330 PRINT"Total dividends"TAB(25)USING"$
    $###,###.##";N*D
340 PRINTUSING"Rate of Return on investm
    ent    ###.##%";(N*D)/C*100
350 GOTO490
360 INPUT"Number of shares";N
370 INPUT"Brokerage fee (percent)";P

```

```

380 IFP=0THENINPUT"Brokerage fee (dollar
    s)";D
390 F=D:IFPTHENF=C*N*P/100
400 RETURN
410 CLS:PRINT
420 PRINTUSING"##### shares at $$,###.##
    # each";N;C
430 IFA=1THENPRINT"Cost";ELSEPRINT"Recei
    ved";
440 PRINT" for shares"TAB(26)USING"$ $###
    ,###.##";C*N
450 PRINT"Brokerage Fee"TAB(26)USING"$ $#
    ##,###.##";F
460 RETURN
470 PRINT"Total Cost for Stock"TAB(26)US
    ING"$ $###,###.##";N*C+F+T
480 RETURN
490 PRINT@282,"Press ENTER to return,"A"
    to repeat";
500 A$=INKEY$:IFVAL(A$)=ATHENCLS:ONAGOTO
    100,160,290
510 IFA$<>CHR$(13)THEN500ELSERETURN

```

Ω



PART II

Business Programs

This section is designed for the businessman, giving him (or her) tools for running and analyzing the business. In today's world, analyzing information about your business and its products make it easier to understand what's happening. How big is your market share? How does product A compare in sales with products B, C and D? Is there a periodic trend to monthly sales? Answers to questions like these can make or break a company.

Several of the programs take advantage of the Model 100's graphics abilities to visually display information in the form of charts.

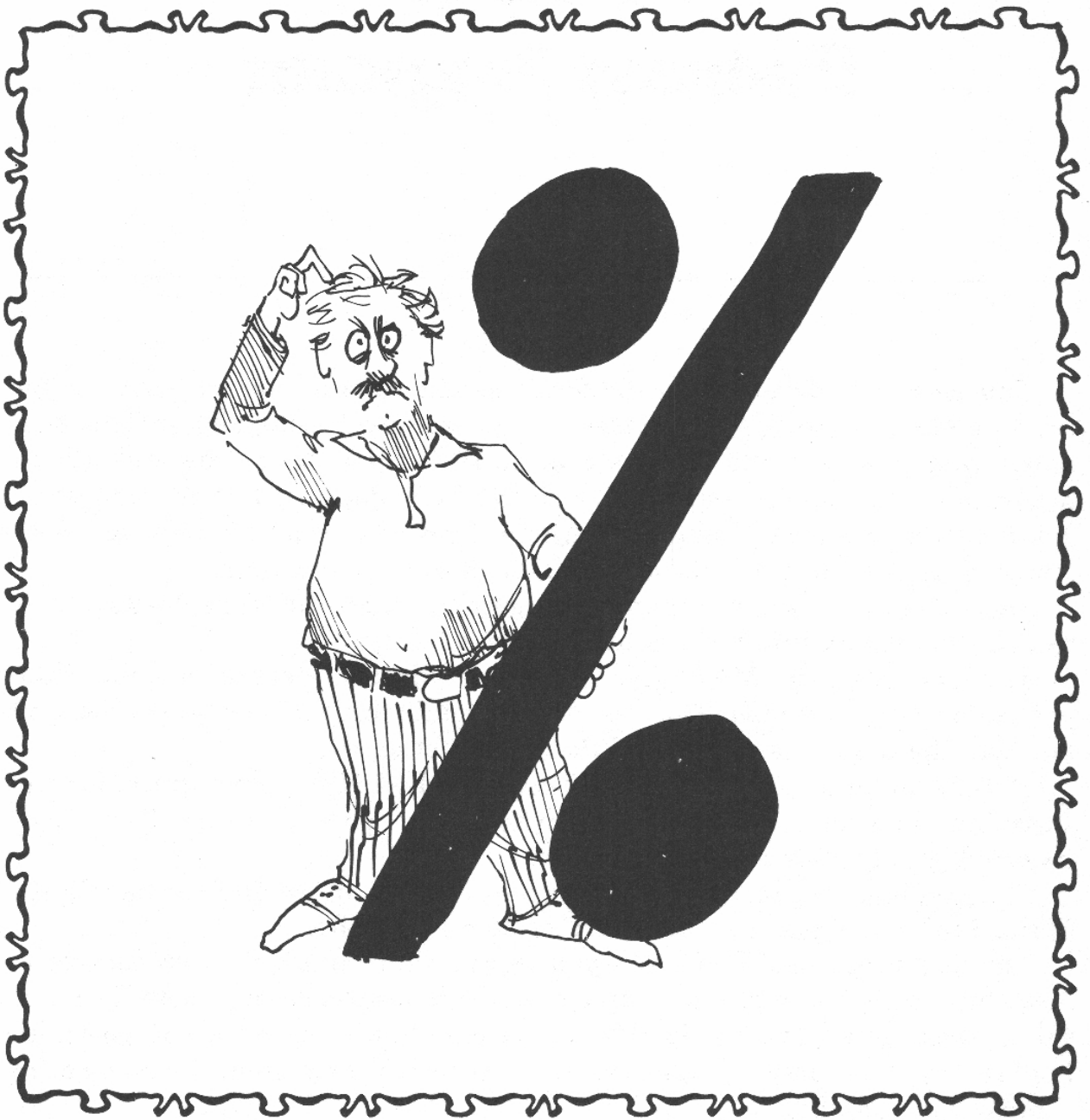
The first program, MARGIN, lets the small businessman determine the percent markup on an item from its cost and retail; find the retail given cost and markup; the retail/cost ratio; and the gross/net profit.

This is followed by an expense account, EXPNSE, a program for the traveling salesman, or anyone else who has to give a record of expenses incurred on a trip for business purposes.

The next four programs are graphic; they display information as bar charts (horizontal and vertical), a pie chart, or as a line chart.

The next program, ZFAC, tells you what's happening with your company, how close to, or far from, bankruptcy your company currently rests. The nice thing about this program is that it becomes more accurate the closer to bankruptcy your company approaches. Important information when an entire company is at stake.

The next three programs, PURCHS, INVOIC and STATEM, let you use the NEBS forms for purchase orders, invoices and statements with your Model 100.



MARGIN

Calculate percent of mark-up.

Knowing how much to markup your merchandise is an important aspect of business. You have to sell the item cheap enough to attract customers, but high enough to stay in business.

The first option in this program lets you find the percentage of markup between the wholesale and retail prices.

And finally, the gross/net profit tells you quickly and simply if you're making enough money to stay in business. You type in your total sales, the cost of the items sold, and your total expenses, and the program tells you what your gross profit is (how much money you earned above the cost of your products), and your net profit (how much money is left after paying rent, taxes, salaries, etc.).

And finally, the gross/net profit tells you quickly and simply if you're making enough money to stay in business. You type in your total sales, the cost of the items sold, and your total expenses, and the program tells you what your gross profit is (how much money you earned above the cost of your products), and your net profit (how much money is left after paying rent, taxes, salaries, etc.).

A very convenient feature of this program is that you don't have to return to the main menu to repeat a function. Pressing the "A" key at the "Press ENTER to return" prompt starts the module over. And you don't have to re-type all the information, just press ENTER for those inputs you want left unchanged, and type new inputs for the others.

Program Listing for MARGIN

```
10 REM 1238 bytes
20 CLS:MAXFILES=0:CLEAR256
30 PRINTTAB(12)"Profit Margins"
```

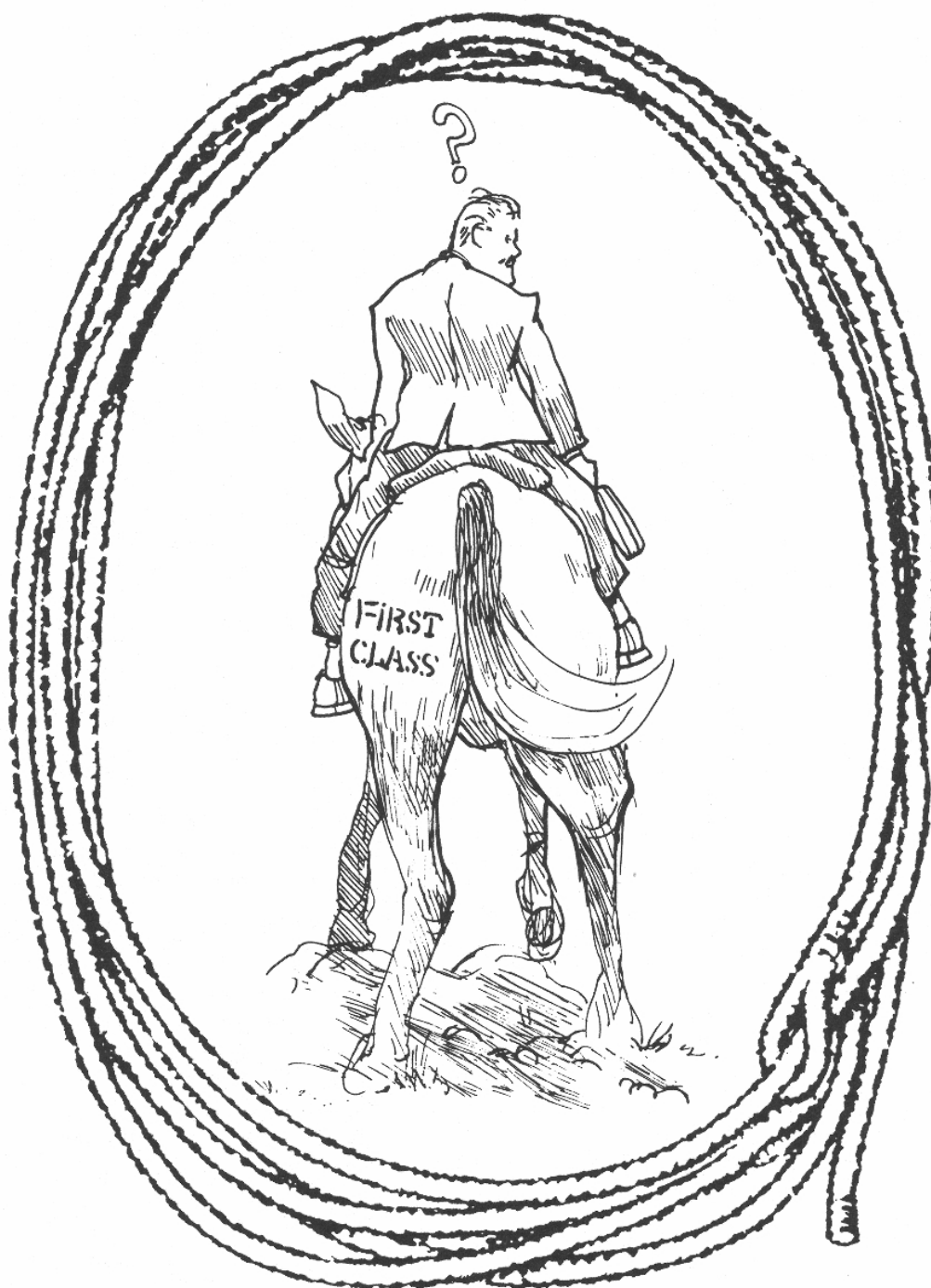
```

40 PRINT:PRINT" 1. Finding Percent Mark
   up"
50 PRINT" 2. Finding Selling Price"
60 PRINT" 3. Retail/Cost ratio (percent
   )"
70 PRINT" 4. Gross/Net Profit"
80 PRINT:PRINTTAB(13)"Your Choice";:INPU
   TA
90 IFA<1ORA>4THEN20
100 ONAGOSUB110,160,210,260:GOTO20
110 CLS:PRINTTAB(10)"Finding Percent Mar
   kup"
120 PRINT:INPUT"Cost of item (dollars)";
   C
130 INPUT"Retail price (dollars)";R
140 PRINT:PRINTUSING"Rate of Markup = ##
   ##.##%";(R-C)/C*100
150 GOTO340
160 CLS:PRINTTAB(10)"Finding Selling Pri
   ce"
170 PRINT:INPUT"Cost of item (dollars)";
   C
180 INPUT"Percent Markup (percent)";P1:P
   =P1/100
190 PRINT:PRINTUSING"Retail Price is $$#
   ##,###.##%";C+(C*P)
200 GOTO340
210 CLS:PRINTTAB(10)"Retail/Cost Ratio"
220 PRINT:INPUT"Cost of item (dollars)";
   C
230 INPUT"Retail price (dollars)";R
240 PRINT:PRINTUSING"Cost is ###.##% of
   Retail price";C/R*100
250 GOTO340
260 CLS:PRINTTAB(10)"Gross/Net Profit"
270 PRINT:INPUT"Total Sales (dollars)";S
280 INPUT"Total Cost of Sales (dollars)"
   ;C
290 INPUT"Total Expenses (dollars)";E
300 PRINTTAB(6)USING"Gross Profits
   $$####.###.##%";S-C
310 PRINTTAB(6)USING"Net Profit
   $$####.###.##%";S-C-E
320 PRINTTAB(6)USING"Gross profits on sa
   les ###.##%";(S-C)/S*100
330 PRINTTAB(6)USING"Net Profit on sales
   ###.##%";(S-C-E)/S*100
340 PRINT@282,"Press ENTER to return,"A"

```

```
to repeat";  
350 A$=INKEY$:IFVAL(A$)=ATHENONAGOTO110,  
160,210,260  
360 IFA$<>CHR$(13)THEN350ELSERETURN
```

Ω



EXPENSE

Keep a record of your business travel expenses.

Anyone who has ever had to prepare an expense account report will appreciate this program.

The key to preparing an expense report is having the receipts to back up the accounting. The problem is organizing and adding up everything into the proper columns for the report, a time consuming activity prone to computational errors.

This program uses a day-by-day approach to its accounting. When you run it, it lists a menu of five items: load data, save data, input data, and list data to either the display or a lineprinter. When you input data you're asked for the cost of each meal, plus any midday snacks. Then you tell it the cost of gas for your car, followed by the cost of any miscellaneous automobile expenses you might have had (oil, tires, etc.). The next prompts ask for the day's beginning and ending mileage, the cost for lodgings for the night, and finally, any miscellaneous expenses that don't fall under any of the previous categories.

The program then goes to the next day's prompt. If you're through entering information, just type -99 and you'll be returned to the main menu where you can save, display or print your data. If you try to load data, you'll erase everything you've just typed.

Saving data is easy; or just press the "2" key and you'll be prompted for the data's filename. Typing in a six-character filename saves the data to memory. If you add the prefix "CAS:" to the filename, the data will be saved to cassette tape.

Loading data is just as simple; just specify the filename and it'll be loaded. Once the data is loaded, you can add more days to it by using the input data option (#3). Don't forget to save the new, longer data file before quitting the program.

The report generated lists the day's total expenses, repeats all the data you've typed in, appropriately labeled of course, and calculates your day's mileage. Each day is printed, then the totals of all the days are printed as the last item.

Currently, the program is limited to ten days, but that can be changed by altering the dimension statement to the number of days you need for your trip.

Program Listing for EXPNSE

```
10 REM 2166 bytes
20 MAXFILES=1: CLEAR256: X=0: DIMD(10,9)
30 CLS: PRINTTAB(10) "Expense Account"
40 PRINT "1. Load data"
50 PRINT "2. Save data"
60 PRINT "3. Input data"
70 PRINT "4. List data"
80 PRINT "5. Print data"
90 PRINTTAB(13) "Your Choice?";
100 A$=INKEY$: IFA$="" THEN 100
110 A=VAL(A$): IFA<1 OR A>5 THEN 100
120 CLS: ON AGOSUB 130,130,230,360,520: GOTO
    30
130 INPUT "Name of file"; F$
140 IFF$="" THEN 130
150 IFA=2 THEN 200
160 OPEN F$ FOR INPUT AS I: PRINT "Loading data"
    "
170 INPUT #I, X: FOR J=1 TO X
180 FOR I=0 TO 9: INPUT #I, D(J,I): NEXT I: NEXT
190 RETURN
200 OPEN F$ FOR OUTPUT AS I: PRINT "Writing dat
    a"
210 PRINT #I, X: FOR J=1 TO X: FOR I=0 TO 9
220 PRINT #I, D(J,I): NEXT I: NEXT I: RETURN
230 CLS: X=X+1: PRINT "day "X" - (enter -99
    to return)"
240 INPUT "Cost of Breakfast"; D(X,0)
250 IF D(X,0)=-99 THEN X=X-1: RETURN
260 INPUT "Cost of Lunch"; D(X,1)
270 INPUT "Cost of Dinner"; D(X,2)
280 INPUT "Cost of snacks"; D(X,3)
290 INPUT "Cost of gas"; D(X,4)
300 INPUT "Miscellaneous car expenses"; D(
    X,5)
310 INPUT "Car beginning mileage"; D(X,6)
320 INPUT "Car ending mileage"; D(X,7)
```

```

330 INPUT"Overnight Lodgings";D(X,8)
340 INPUT"Miscellaneous expenses";D(X,9)
350 GOTO230
360 FORI=1TOX:GOSUB380:NEXT
370 I=0:GOSUB380:RETURN
380 CLS:IFI>0THENPRINT"Day"I;;ELSEPRINTX
    "days.";
390 PRINTTAB(15)"Total expenses $";
400 A=0:FORJ=0TO9:IFJ<>6ANDJ<>7THENA=A+D
    (I,J):IFI<>0THEND(0,J)=D(0,J)+D(I,J
    )
410 NEXT:PRINTA
420 PRINT"Meals - $"D(I,0)+D(I,1)+D(I,2)
    +D(I,3)
430 PRINTUSING" B-$$$$.## L-$$$$.## D-$$$
    ##.## S-$$$$.##";D(I,0);D(I,1);D(I,
    2);D(I,3);
440 PRINT"Gas $"D(I,4)TAB(13)"Misc. car
    expenses $"D(I,5)
450 IFI<>0THENA=D(I,7)-D(I,6):D(0,7)=D(0
    ,7)+A
460 PRINT"Miles driven";:IFI=0THENPRINTD
    (0,7)ELSEPRINTA
470 IFI<>0THENPRINT" (Start-"D(I,6)"End-
    "D(I,7)""")
480 PRINT"Lodgings $";D(I,8)TAB(20);"Mis
    c. exp. $";D(I,9)
490 GOTO500
500 PRINT@287,"Press ENTER to continue";
510 IFINKEY$<>CHR$(13)THEN510ELSERETURN
520 LPRINT"Expense Account Report":FORI=
    1TOX:GOSUB540:NEXT
530 I=0:GOSUB540:LPRINTCHR$(12):RETURN
540 CLS:PRINT"Sending data to lineprinte
    r."
550 IFI>0THENLPRINT"Day"I;;ELSELPRINTX"d
    ays.";
560 LPRINTTAB(15)"Total expenses $";
570 A=0:FORJ=0TO9:IFJ<>6ANDJ<>7THENA=A+D
    (I,J):IFI<>0THEND(0,J)=D(0,J)+D(I,J
    )
580 NEXT:LPRINTA
590 LPRINT"Meals - $"D(I,0)+D(I,1)+D(I,2)
    +D(I,3)
600 LPRINTUSING" Breakfast -$$$$.##; L
    unch -$$$$.##; Dinner -$$$$.##; S
    nacks -$$$$.##";D(I,0);D(I,1);D(I,2
    );D(I,3)

```

```

610 LPRINT"Gas $"D(I,4)TAB(30)"Miscellan
    eous car expenses $"D(I,5)
620 IFI<>0THENA=D(I,7)-D(I,6):D(0,7)=D(0
    ,7)+A
630 LPRINT"Miles driven";:IFI=0THENLPRIN
    TD(0,7)ELSELPRINTA
640 IFI<>0THENLPRINT"  (Starting Mileage
    -"D(I,6)TAB(30)"Ending mileage -"D
    (I,7)""
650 LPRINT"Overnight lodgings $";D(I,8)T
    AB(30);"Miscellaneous expenses $";D
    (I,9)
660 LPRINT:LPRINTSTRING$(75,95):LPRINT:R
    ETURN

```

Ω

BARV

A vertical bar chart graphic program.

This program generates a vertical bar chart of five items. That is, the names of the items are located across the bottom of the display, while the numbers indicating the quantities are at the left side of the display. A bar starts over each item name, and goes up towards the top of the display.

The names (labels) of the items are limited to five characters each.

To use the program, type in the name of an item in response to the "Label?" prompt, then type in the value. If you have less than five items, press the ENTER key in answer to the label prompt, otherwise the program automatically continues when you've typed in the fifth item.

The bars can be drawn with either the lowest number you input as the starting value for the bars, or with zero as the starting value. Thus, if your highest value is 1000 and the lowest is 900, choosing the zero option will have the lowest number zero, the next number will be 166, the next number 332, and so forth until 1000 is reached at the top. The shortest bar displayed will run from zero to 900. Choosing the other option will put the 900 at the bottom, with 916 as the next value, and so forth until 1000. The lowest bar will be at the bottom of the display, opposite the 900.

If you want, you can redraw the same display, or do another graph.

Program Listing for BARV

```
10 REM 1245 bytes
20 MAXFILES=0: CLEAR256:CLS
30 PRINTTAB(10)"Vertical Bar Chart"
40 PRINT"Enter labels and amount (five l
   etters"
50 PRINT"maximum, five labels maximum)"
```

```

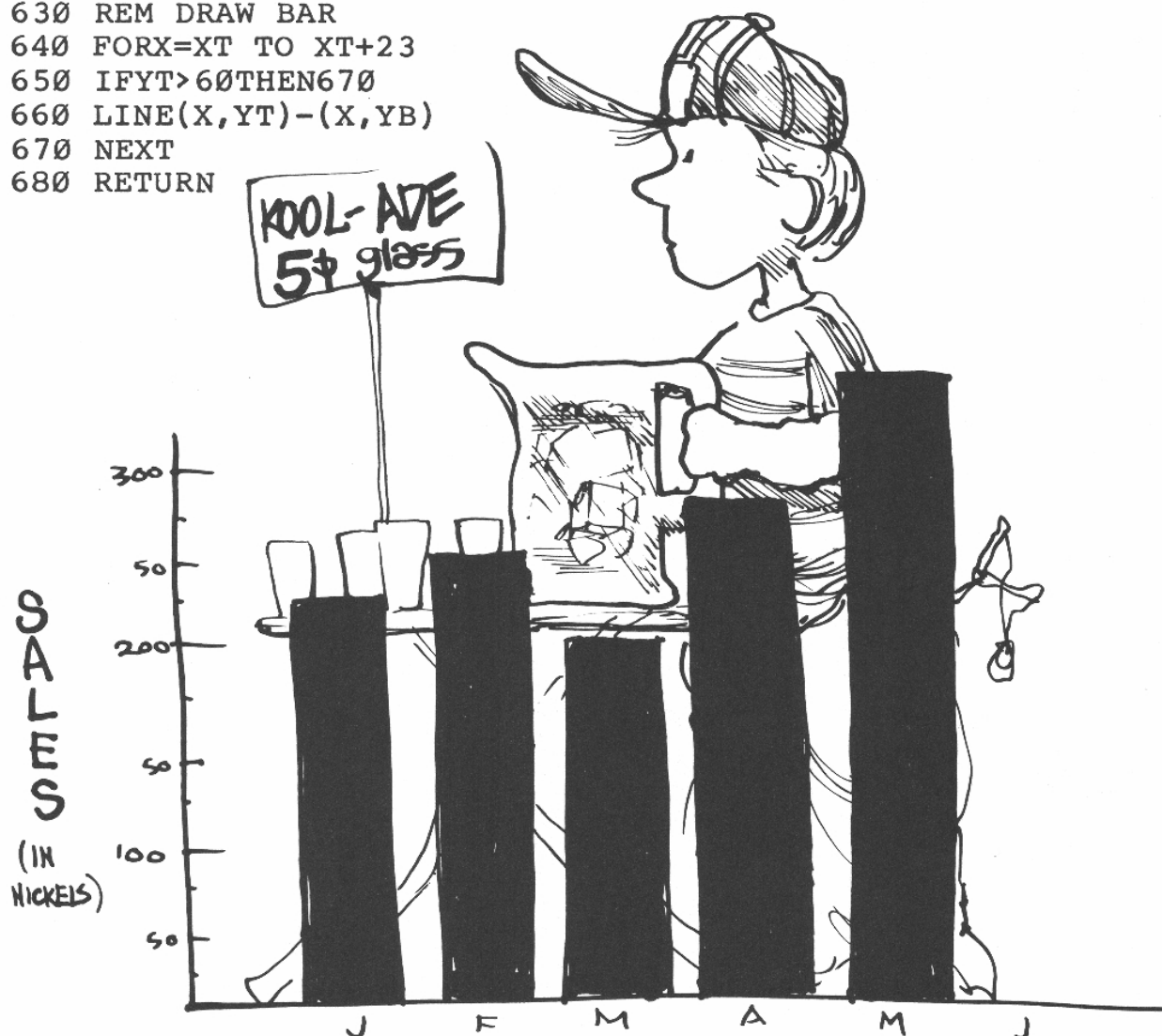
60 PRINT"Enter null to end input"
70 DIM L$(5),A(5),L(8),P(5)
80 PO=0
90 FORX=1TO8:L(X)=PO:PO=PO+40:NEXT
100 N=1
110 PRINT"Label #";N;
120 INPUTL$(N)
130 IFL$(N)=""THENN=N-1:GOTO190
140 IFLEN(L$(N))<4THENL$(N)=STRING$((5-LEN(L$(N)))/2,32)+L$(N)
150 L$(N)=LEFT$(L$(N),5)
160 INPUT"Amount";A(N)
170 N=N+1:IFN>5THENN=N-1:GOTO190
180 GOTO110
190 NH=0:NL=32767
200 FORX=1TON
210 IFA(X)>NHTHEN NH=A(X)
220 IFA(X)<NLTHEN NL=A(X)
230 NEXT:CLS
240 PRINT"Enter zero for full scale, 1 f
    or default";
250 PRINT"(default is lowest number,"NL"
    )"
260 INPUTQ:IFQ=0THENNL=0
270 CLS
280 DI=NH-NL
290 LI=DI/6
300 I=NL
310 FORX=7TO1STEP-1
320 PRINT@L(X),USING"#####";I;
330 I=I+LI
340 NEXT
350 PO=L(8)+7
360 FORX=1TON
370 PRINT@PO,L$(X);
380 PO=PO+6
390 NEXT
400 YB=55
410 FORL=1TON
420 P(L)=((A(L)-NL)*100)/(NH-NL)
430 NEXT
440 FORL=1TON
450 XT=(L*36)+9
460 YT=55-((P(L)/100)*55)
470 GOSUB640
480 NEXT
490 A$=INKEY$:IFA$=""THEN490
500 CLS:PRINT

```

```

510 PRINT"Another chart (Y/N)?"
520 A$=INKEY$:IFA$=""THEN520
530 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
540 IFA$="Y"THEN570
550 IFA$<>"N"THEN520
560 END
570 PRINT"Redraw the same chart (Y/N)?"
580 A$=INKEY$:IFA$=""THEN580
590 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
600 IFA$="N"THEN RUN
610 IFA$="Y"THEN190
620 GOTO580
630 REM DRAW BAR
640 FORX=XT TO XT+23
650 IFYT>60THEN670
660 LINE(X,YT)-(X,YB)
670 NEXT
680 RETURN

```



Ω

BARH

A horizontal bar chart graphic program.

This program is the same as the previous, except that the bars are oriented horizontally across the display instead of vertically, the labels are on the left side of the display, and the numbers are across the bottom. On this chart there's room for seven labels. Other than these changes, the same restrictions and options apply to this program as to the other.

Program Listing for BARH

```
10 REM 1177 bytes
20 MAXFILES=0: CLEAR256: CLS
30 PRINTTAB(10); "Horizontal Bar Chart"
40 PRINT "Enter labels and amount five le
   tters."
50 PRINT "(Maximum SIX labels.)"
60 PRINT "Enter null to end input."
70 DIM L$(7), A(7), L(8), P(7)
80 PO=0
90 FORX=1 TO 8: L(X)=PO: PO=PO+40: NEXT
100 N=1
110 PRINT "Label #"; N;
120 INPUT L$(N)
130 IF L$(N)="" THEN N=N-1: GOTO 190
140 L$(N)=STRING$(5, 32)+L$(N)
150 L$(N)=RIGHT$(L$(N), 5)
160 INPUT "Amount "; A(N)
170 N=N+1: IF N>7 THEN N=N-1: GOTO 190
180 GOTO 110
190 NH=0: NL=32767
200 FORX=1 TO N
```

```

210 IFA(X)>NH THEN NH=A(X)
220 IFA(X)<NL THEN NL=A(X)
230 NEXT:CLS
240 PRINT"Enter 0 for full scale, enter
    1 for"
250 PRINT"default to lowest number";NL
260 INPUTQ:IFQ=0 THEN NL=0
270 CLS
280 DI=NH-NL
290 LI=DI/5
300 I=NL
310 PO=L(8)+5
320 FORX=1 TO 7
330 IFX<=N THEN PRINT@L(X),L$(X);
340 IFX<7 THEN PRINT@PO-1,I;
350 I=I+LI
360 PO=PO+6
370 NEXT
380 FORL=1 TO N
390 P(L)=((A(L)-NL)*100)/(NH-NL)
400 NEXT
410 FORL=1 TO N
420 Y=(L-1)*8
430 X=((P(L)/100)*187)+30
440 GOSUB610
450 NEXT
460 A$=INKEY$:IFA$="" THEN 460
470 CLS:PRINT
480 PRINT"Another chart (Y/N)?"
490 A$=INKEY$:IFA$="" THEN 490
500 IF ASC(A$)>90 THEN A$=CHR$(ASC(A$)-32)
510 IFA$="Y" THEN 540
520 IFA$<>"N" THEN 490
530 END
540 PRINT"Redraw the same chart (Y/N)?"
550 A$=INKEY$:IFA$="" THEN 550
560 IF ASC(A$)>90 THEN A$=CHR$(ASC(A$)-32)
570 IFA$="N" THEN RUN
580 IFA$="Y" THEN 190
590 GOTO 550
600 REM DRAW BARS
610 FOR YT=Y TO Y+5
620 IF YT>60 THEN 640
630 LINE(30,YT)-(X,YT)
640 NEXT
650 RETURN

```

Ω

PIE

A pie chart graphic program.

Another useful chart is the pie chart, which graphically illustrates how the money pie is divided up.

In this program you can have up to eight assigned pieces in the pie. You type in the name of the item, up to nine characters in length, and then the value assigned to that label. When you're finished, press ENTER at the "Label?" prompt. If you have eight items, the program automatically continues after the eighth entry.

After organizing the items into a lowest-number-first order you're asked for a total. This number should be the total of the values of the different items, but it doesn't have to be, it can be larger. Then the items are listed on the left of the display, and a letter is assigned to each label. The pie chart is drawn and divided up into sections. The sections are apportioned by calculating the percentage of the total of each item. As this is done, the items' numbers are added together. Each section has a letter beside it to indicate to which label it belongs. After drawing in and labeling the pie sections, the program compares their total with the TOTAL figure you typed in earlier. If they are different, the label "Other" is placed on the section of the pie leftover to indicate a remainder unaccounted for in the total of the items.

Program Listing for PIE

```
10 REM 1043 bytes
20 CLS: CLEAR 256: MAXFILES=0
30 DIM V(8), L$(8), P(8)
40 PRINT TAB(10) "Pie Chart program"
50 PRINT "Enter the label, and its value"
```

```

60 PRINT"Null for label when done"
70 N=1
80 INPUT"Label";L$(N)
90 IFL$(N)=" "THENN=N-1:GOTO120
100 INPUT"Value";V(N)
110 N=N+1:IFN<9THEN80
120 TT=0:FORL=1TON:TT=TT+V(L):NEXT
130 INPUT"TOTAL ";T
140 IFT<TTTHEN T=TT
150 C=0:FORX=1TON-1
160 IFV(X)<=V(X+1)THEN220
170 A=V(X):B=V(X+1)
180 V(X)=B:V(X+1)=A
190 A$=L$(X):B$=L$(X+1)
200 L$(X)=B$:L$(X+1)=A$
210 C=1
220 NEXT
230 IFC=1THEN150
240 FORX=1TON
250 P(X)=(V(X)*100)/T
260 NEXT
270 CLS
280 FORX=1TON
290 PRINTCHR$(64+X);" ";
300 PRINTUSING"\          \#####.# ##.###"
    ;L$(X);V(X);P(X)
310 NEXT
320 D=3.14159/180
330 XC=195:YC=32
340 FORA=0TO360STEP2
350 C=COS(A*D)*32
360 S=SIN(A*D)*32
370 C=C+XC:S=S+YC
380 PSET(C,S):NEXT
390 REM DRAW PIE
400 AT=0:FORX=0TON
410 A=(P(X)*360)/100
420 AT=AT+A
430 C=(COS(AT*D)*32)+XC
440 S=(SIN(AT*D)*32)+YC
450 LINE(XC,YC)-(C,S)
460 IFX=0THEN500
470 R=C/6+(INT(S/8))*40
480 IFC<185THENPRINT@R-1,CHR$(64+X);
490 IFC>184THENPRINT@R+1,CHR$(64+X);
500 NEXT
510 IFAT<359THENPRINT@155,"Other";
520 IFINKEY$=""THEN520

```

```
530 PRINT@280,STR$(27,32);  
540 PRINT@280,"Another chart?";  
550 A$=INKEY$:IFA$="Y"ORA$="y"THEN30  
560 END
```



Ω

LINE

A graphing program using lines.

This graphing program assumes you want to trace the performance of a particular item over a period of time, thirty days, weeks, or perhaps months.

Unlike the previous three programs, this one doesn't ask you for the labels, instead it asks you for the values. Keep typing in values in response to the prompt until you've entered them all (thirty maximum). If you have less than thirty, type the "=" key in response to the prompt, and the program will go to the next step.

Once the highest and lowest values are determined, you're then asked if you want to use zero as the lowest value, or use the lowest number you typed in as the bottom figure.

The graph is drawn with the numbers at the left of the display, with the numbers one to thirty, in steps of five, displayed across the bottom line. Then the lines for the various values are drawn, in the order you typed them in. This lets you see how the values have risen and fallen over time. If the numbers are weekly sales in dollars, you can see how your business has cycled for the last thirty-two weeks. The numbers could just as easily be hourly, daily, monthly, or even yearly.

When you've finished, you can redraw the same graph, or input data for a new one.

Program Listing for LINE

```
10 REM 1176 bytes
20 MAXFILES=0: CLEAR 256: CLS
30 PRINT TAB(10) "Line Graph Chart"
40 PRINT "Enter amounts. Up to 32 may be
    entered."
```

```

50 PRINT"Enter = to end input"
60 DIMA(30),L(8),P(32)
70 PO=0
80 FORX=1TO8:L(X)=PO:PO=PO+40:NEXT
90 N=1
100 PRINTUSING"Amount ##";N;
110 INPUTL$:IFL$="="THENN=N-1:GOTO160
120 IFL$=""THEN100
130 A(N)=VAL(L$)
140 N=N+1:IFN>30THENN=N-1:GOTO160
150 GOTO100
160 NH=0:NL=32767:PRINT"Finding highest
    and lowest values."
170 FORX=1TON
180 IFA(X)>NHTHEN NH=A(X)
190 IFA(X)<NLTHEN NL=A(X)
200 NEXT:CLS:Q=1
210 PRINT"Enter zero for full scale, 1 f
    or default";
220 PRINT"(default is lowest number,"NL"
    )"
230 INPUTQ:IFQ=0THENNL=0
240 CLS
250 DI=NH-NL
260 LI=DI/6
270 I=NL
280 FORX=7TO1STEP-1
290 PRINT@L(X),USING"#####";I;
300 I=I+LI
310 NEXT:PRINT@280,"Period:1    5    10
    15    20    25    30";
320 LINE(39,0)-(39,54):LINE(37,54)-(239,
    54)
330 YB=53
340 FORL=1TON
350 P(L)=((A(L)-NL)*100)/(NH-NL)
360 NEXT
370 FORL=1TON
380 XT=(L*6)+38
390 YT=53-((P(L)/100)*53)
400 IFYT>60THEN420
410 LINE(XT,YB)-(XT,YT)
420 NEXT
430 A$=INKEY$:IFA$=""THEN430
440 CLS:PRINT
450 PRINT"Redraw the same chart (Y/N)?"
460 A$=INKEY$:IFA$=""THEN460
470 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)

```

```
480 IFA$="Y"ORA$="y"THEN160
490 IFA$<>"N"ANDA$<>"n"THEN460
500 PRINT"Another chart (Y/N)?"
510 A$=INKEY$:IFA$=""THEN510
520 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
530 IFA$="Y"ORA$="y"THEN20
540 IFA$<>"N"ANDA$<>"n"THEN510
550 END
```

Ω

ZFAC

Determine the health of your business.

The Z-Factor is an equation which tells you how your business is faring in the real world, whether it's solvent or approaching bankruptcy.

A number of years ago Mr. Edward I. Altman developed this formula to determine the financial status of a business. The formula is often used by stockbrokers to determine whether a company is a good investment.

According to Mr. Altman's tests, the formula becomes more accurate as a company gets closer to bankruptcy.

The most extensive tests with this formula have been with businesses having assets between one million and twenty-five million dollars.

Mr. Altman has said that keeping an eye on the Z-Factor of your business, as well as other indicators, can be helpful in managing your business.

Generally, the Z-Factor for a business will be a number between -4 and +8. The higher the number the better your business's situation. A value of 1.81 or less denotes a severe financial crisis. Values 2.99 or above show financial strength. Numbers in between these two limits indicates a business which could go either way, success or failure.

Once you've determined your company's Z-Factor, you may want to experiment with the effect of increasing or decreasing some of the factors. To do this experimentation, use the up and down arrows to move the cursor to the value you want to change, then type in the new number. Use the back-arrow to correct mistakes. When the number is completed, press ENTER, and the new Z-Factor will be computed and displayed.

You may also wish to work the formula backwards, for example: "What do my total assets need to be for a Z-Factor of 2.99?"

To do this type of calculation, move the cursor to the Z-Factor line, type in the desired Z-Factor value, and press ENTER. Now use the arrow keys to move the

cursor to the line you want computed (Total Assets, in this example) and press ENTER. The new values will be quickly computed.

For more information about the Z-Factor, read "Corporate Bankruptcy" by Edward I. Altman, published by D.C. Heath, or "Corporate Collapse: the causes and symptoms" by John Argenti, published by John Wiley and Sons.

To experiment with the program, you can enter "X" in response to the "Working capital?" prompt and a dummy company, using \$5,555 for all the values, will be displayed.

Also, once you've entered all the data on your company, you can save the information to memory or cassette tape by typing an "S" while in the Z-Factor display. Similarly, you can send a copy of the display to your printer by typing a "P." If you've already saved data on a company and you want to load it back, you can type an "L" either in response to the beginning "Working capital?" prompt, or while in the Z-Factor display. Finally, you can restart the program from the Z-Factor display by typing an "X."

Program Listing for ZFAC

```
10 REM 3167 bytes
20 CLEAR256:CLS
30 PRINTTAB(15)"Z Factor"
40 DEFSNGT,W,R,E,M,T,S,Z,N
50 PRINT"Please enter the following information."
60 B$="$ $#####.###.##"
70 INPUT"Working capital";WC$:IFWC$=""OR
   WC$="0"THEN70ELSEIFWC$="X"ORWC$="x"
   THEN170
80 IF WC$="L"ORWC$="1" THEN820
90 WC=VAL(WC$)
100 INPUT"Total Assets";TA:IFTA=0THEN100
110 INPUT"Retained earnings";RE:IF RE=0THEN110
120 INPUT"Earnings before interest and taxes";EB:IF EB=0THEN120
130 INPUT"Market value of equity";ME:IF ME=0THEN130
140 INPUT"Total Debt book value";TD:IF TD=0THEN140
150 INPUT"Sales dollars";SA:IF SA=0THEN150
160 GOTO190
170 XX=5555:TA=XX:WC=XX:RE=XX:EB=XX:ME=XX:TD=XX:SA=XX
180 CLS
```

```

190 REM Print data on screen
200 PRINT@0,"Total assets"TAB(25)USINGB$
    ;TA
210 PRINT"Working capital"TAB(25)USINGB$
    ;WC
220 PRINT"Retained earnings"TAB(25)USING
    B$;RE
230 PRINT"Earn. before int.& taxes"TAB(2
    5)USINGB$;EB
240 PRINT"Equity market value"TAB(25)USI
    NGB$;ME
250 PRINT"Book value of debt"TAB(25)USIN
    GB$;TD
260 PRINT"Sales"TAB(25)USINGB$;SA
270 IFZZ=1THEN290
280 Z=1.2*(WC/TA)+1.4*(RE/TA)+3.3*(EB/TA
    )+.6*(ME/TD)+(SA/TA)
290 PRINT"Z Factor"TAB(32)USING"####.##"
    ;Z;
300 IFZZ=1THEN RETURN
310 REM MOVE CURSOR
320 IFAA=0THENP=6*40+25:AA=1
330 PRINT@P,CHR$(95);
340 I$=INKEY$:IFI$=""THEN340
350 IFZZ=1THENIFASC(I$)=13THENRETURNELSE
    GOTO410:REM CHECKING FOR Z
360 IFI$="S"ORI$="s"THEN870
370 IFI$="L"ORI$="L"THEN820
380 IFI$="P"ORI$="P"THEN960
390 IFI$="X"ORI$="x"THEN20
400 IFASC(I$)<58ANDASC(I$)>44THENNO$=""
    :PL=P+1:PRINT@PL,"";:G
    OSUB480
410 IFASC(I$)=30THENP=P-40:IFP<25THENP=2
    5:ELSEGOSUB530
420 IFASC(I$)=31THENP=P+40:IFP>7*40+25TH
    ENP=P-40:ELSEGOSUB520
430 GOTO330
440 REM'ENTERING A NUMBER
450 NO$="":PL=P+1
460 I$=INKEY$:IFI$=""THEN460
470 IF(ASC(I$)=8ORASC(I$)=13)ANDNO$=""TH
    EN460
480 IFASC(I$)<58ANDASC(I$)>44THENPRINT@P
    L,I$;:PL=PL+1:NO$=NO$+I$:I$="":GOTO
    460
490 IFASC(I$)=13THENNO=VAL(NO$):IFNO=0TH
    ENI$=CHR$(8):GOTO500:ELSE550

```



```

500 IFASC(I$)=8THENNO$(1)=LEFT$(NO$(LEN
    (NO$)-1)):NO$=NO$(1):PL=PL-1:PRINT@
    P+1,NO$;" ";:GOTO460
510 GOTO460
520 PRINT@P-40," ";:RETURN
530 PRINT@P+40," ";:RETURN
540 REM FIND CURSOR POSITION & CHANGE VA
    RIABLE
550 PL=P:X=0
560 PL=PL-40:X=X+1:IFPL>0THEN560
570 IFZZ=1THENRETURN: REM CHECKING FOR Z
580 ONXGOSUB600,610,620,630,640,410,660,
    670
590 GOTO190
600 TA=NO:RETURN
610 WC=NO:RETURN
620 RE=NO:RETURN
630 EB=NO:RETURN
640 ME=NO:RETURN
650 TD=NO:RETURN
660 SA=NO:RETURN
670 Z=NO:GOTO690
680 REM GET VARIABLE AND BACKFIGURE Z
690 ZZ=1:GOSUB190
700 PRINT@280," Put CURSOR on factor,
    press ENTER. ";
710 AA=0:GOSUB320:GOSUB550
720 ONXGOSUB750,760,770,780,790,800,810
730 ZZ=0
740 GOTO190
750 TA=(1.2*WC*TD+1.4*RE*TD+3.3*EB*TD+SA
    *TD)/(Z*TD-.6*ME):RETURN
760 WC=-1*(1.4*RE+3.3*EB+ (.6*ME*TA)/TD+S
    A-Z*TA)/1.2:RETURN
770 RE=-1*(1.2*WC+3.3*EB+ (.6*ME*TA)/TD+S
    A-TA*Z)/1.4:RETURN
780 EB=-1*(1.2*WC+ (.6*ME*TA)/TD+1.4*RE+S
    A-TA*Z)/3.3:RETURN
790 ME=TD*(1.2*WC/TA+1.4*RE/TA+3.3*EB/TA
    +SA/TA-Z)/-.6:RETURN
800 TD=-.6*ME/(1.2*WC/TA+1.4*RE/TA+3.3*E
    B/TA+SA/TA-Z):RETURN
810 SA=-1*(1.2*WC/TA+1.4*RE/TA+3.3*EB/TA
    + (.6*ME/TD)-Z)*TA:RETURN
820 CLS:INPUT"Filename";FI$
830 OPENFI$FORINPUTAS1
840 INPUT#1,TA,WC,RE,EB,ME,TD,SA
850 CLOSE

```

```

860 GOTO180
870 GOSUB940
880 CLS:INPUT"Filename";FI$
890 OPENFI$FOROUTPUTAS1
900 PRINT"Saving ";FI$
910 PRINT#1,TA;WC;RE;EB;ME;TD;SA
920 CLOSE
930 PRINT"Data saved":FORX=1TO1500:NEXT:
    GOTO180
940 FI$="":PS$="":DR$="":SP$="":RETURN
950 RETURN
960 REM LPRINT
970 CLS:PRINT"Is the lineprinter ready?"
980 I$=INKEY$:IFI$=""THEN980
990 IFI$="N"THEN180
1000 IFI$<>"Y"THEN970
1010 LPRINT"TOTAL ASSETS"TAB(40)USINGB$;
    TA
1020 LPRINT"WORKING CAPITAL"TAB(40)USING
    B$;WC
1030 LPRINT"RETAINED EARNINGS"TAB(40)USI
    NGB$;RE
1040 LPRINT"EARNINGS BEFORE INTEREST AND
    TAXES"TAB(40)USINGB$;EB
1050 LPRINT"MARKET VALUE OF EQUITY"TAB(4
    0)USINGB$;ME
1060 LPRINT"BOOK VALUE OF DEBT"TAB(40)US
    INGB$;TD
1070 LPRINT"SALES DOLLARS"TAB(40)USINGB$
    ;SA
1080 LPRINT
1090 LPRINT"Z FACTOR"TAB(40)USINGB$;Z
1100 LPRINT:LPRINT:LPRINT:LPRINT:GOTO180

```

Ω

PURCHS, INVOICE, STATEM

Create purchase orders, invoices and monthly statements.

One of the advantages of a computer is the capability of using preprinted, tractor-feed forms to relieve much of the tedium of filling out forms, as well as reducing the odds of computational errors.

These programs use the NEBS forms (New England Business Systems) for generating purchase orders (#9055), invoices (#9040), and statements (#9060), automatically filling in the proper spacing and blanks. You give the destination, date, numbers quantities, descriptions, unit costs and other information, and the programs fill out the forms, including computing the extended costs and total expense for the order/statement.

When you're finished, all you do is separate the three-part form before mailing it. If you want, you could add a simple tape-save routine that would give you a record of every form sent. A load routine would let you make as many copies of a form as you wanted, in addition to being able to verify what the form actually said in the event of a dispute.

For a catalog of NEBS forms, write: NEBS, 12 South St., Townsend, MA 01469.

If you have another supplier you would rather use, just modify the programs to match the form you want to use.

Program Listing for PURCHS

```
10 REM 1430 bytes
20 CLS
30 CLEAR350
40 DATA"Your Company Name"
50 DATA"Your Street Address"
60 DATA"Town, State, and Zip"
```

```

70 DATA"Your Phone Number"
80 DIM C$(3),N$(3),D$(34),Q$(34),I(34),U
   C(34)
90 L=0
100 FORX=1TO3
110 READ C$(X):IFLEN(C$(X))>LTHENL=LEN(C
   $(X))
120 NEXT
130 READPH$
140 FORX=1TO3:IFLEN(C$(X))=LTHEN170
150 DI=(L-LEN(C$(X)))/2
160 C$(X)=STRING$(DI,32)+C$(X)
170 NEXT
180 PRINT"Purchase Order Form for:":PRIN
   T
190 FORX=1TO3:PRINTTAB(10);C$(X)
200 NEXT
210 PRINT"Company P.O. is going to?"
220 LINEINPUTN$(1)
230 PRINT"Street address?"
240 LINEINPUTN$(2)
250 PRINT"Town, State, and Zip code?"
260 LINEINPUTN$(3)
270 L=0:FORX=1TO3
280 IFLEN(N$(X))>LTHENL=LEN(N$(X))
290 NEXT
300 FORX=1TO3
310 IFLEN(N$(X))=LTHEN340
320 DI=(L-LEN(N$(X)))/2
330 N$(X)=STRING$(DI,32)+N$(X)
340 NEXT
350 CLS:PRINT"Purchase Order id going to
   :":PRINT
360 FORX=1TO3:PRINTTAB(10);N$(X):NEXT
370 INPUT"Purchase Order number";PO$
380 INPUT>Date for Purchase Order";D$
390 PRINT:PRINT"34 items maximum, enter
   Null to end."
400 N=1
410 PRINT"Description for item"N"?":LIN
   EINPUTD$(N)
420 IFD$(N)=" "THENN=N-1:GOTO480
430 INPUT"Quantity required";Q(N)
440 INPUT"Item number";I(N)
450 INPUT"Unit cost";UC(N)
460 N=N+1:IFN>34THENN=N-1:GOTO480
470 GOTO410
480 CLS:PRINT"Ready printer and press EN

```

```

    TER."
490 A$=INKEY$:IFA$<>CHR$(13)THEN490
500 VT=3:GOSUB720
510 LPRINTTAB(74);PO$
520 PRINT"Does your form have your Compa
    ny    name on it (Y/N)?"
530 A$=INKEY$:IFA$=" "THEN530
540 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
550 IFA$="Y"THENVT=8:GOTO590
560 FORX=1TO3:LPRINTTAB(18);C$(X)
570 NEXT
580 VT=4
590 GOSUB720
600 FORX=1TO3
610 LPRINTTAB(20);N$(X)
620 NEXT
630 VT=10:GOSUB720
640 FORX=1TON
650 LPRINTUSING" #####";Q(X);
660 LPRINTTAB(18);I(X);
670 LPRINTTAB(34);D$(X);
680 LPRINTTAB(66);UC(X);
690 LPRINTTAB(79);Q(X)*UC(X)
700 NEXT
710 END
720 FORV=1TOVT
730 LPRINT
740 NEXT:RETURN

```

Program Listing for INVOIC

```

10 REM 1981 bytes
20 CLS
30 CLEAR350
40 DATA"Name of your Company
50 DATA"Your stree address
60 DATA"Town, State  Zip
70 DATA"(603) 555-1212
80 DIM C$(3),N$(3),D$(17),Q(17),UP(17)
90 L=0
100 FORX=1TO3
110 READ C$(X):IFLEN(C$(X))>LTHENL=LEN(C
    $(X))
120 NEXT
130 READ YP$
140 FORX=1TO3:IFLEN(C$(X))=LTHEN170

```

```

150 DI=(L-LEN(C$(X)))/2
160 C$(X)=STRING$(DI,32)+C$(X)
170 NEXT
180 PRINT"Invoice Form for":PRINT
190 FORX=1TO3:PRINTTAB(10);C$(X)
200 NEXT
210 PRINT"Company you're shipping to?";
220 LINEINPUTN$(1)
230 LINEINPUT"Street Address?";N$(2)
240 LINEINPUT"Town, State, and Zip?";N$(
    3)
250 L=0:FORX=1TO3
260 IFLEN(N$(X))>LTHENL=LEN(N$(X))
270 NEXT
280 FORX=1TO3
290 IFLEN(N$(X))=LTHEN320
300 DI=(L-LEN(N$(X)))/2
310 N$(X)=STRING$(DI,32)+N$(X)
320 NEXT
330 CLS:PRINT"Invoice is going to:":PRIN
    T
340 FORX=1TO3:PRINTTAB(10);N$(X):NEXT
350 INPUT"Invoice control number";IC$
360 INPUT"Date shipped";DS$
370 INPUT"Account Number";AN$
380 INPUT"Date of invoice";DI$
390 CLS:PRINT"Shipping instructions"
400 PRINT"(up to five lines long, Null t
    o end)."
```

```

410 FORX=1TO5
420 SH$(X)=""
430 PRINT"Enter line #";X;"?":LINEINPUTS
    H$(X)
440 IFSH$(X)=""THENX=5
450 NEXT
460 CLS:PRINT"Input items to ship, Null
    when done."
470 PRINT"Maximum of 17 different items.
    "
480 N=1
490 PRINT:PRINT:LINEINPUT"Description?";
    D$(N)
500 IFD$(N)=""THENN=N+1:GOTO540
510 INPUT"Quantity";Q(N)
520 INPUT"Unit price";UP(N)
530 N=N+1:GOTO490
540 CLS:PRINT"Start printer output. Set
    your printer"
```

```

550 PRINT"to the form's top, and press E
    NTER."
560 A$=INKEY$:IFA$<>CHR$(13)THEN560
570 CLS
580 VT=3:PRINT"Do your forms have the na
    me of your"
590 PRINT"company preprinted on them (Y/
    N)?"
600 A$=INKEY$:IFA$=""THEN600
610 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
620 IFA$<>"Y"ANDAS$<>"N"THEN600
630 IFA$="Y"THENV=3:GOSUB930:GOTO670
640 FORX=1TO3:LPRINTTAB(11);C$(X);
650 IFX<>3THENLPRINT
660 NEXT
670 LPRINTTAB(50);IC$
680 LPRINT:LPRINT
690 IFA$="N"THENLPRINTTAB(10);YP$;
700 LPRINTTAB(67);DI$
710 VT=3:GOSUB930
720 FORX=1TO5
730 IFX>3THEN750
740 LPRINTTAB(11);N$(X);
750 LPRINTTAB(50);SH$(X)
760 NEXT
770 VT=3:GOSUB930
780 LPRINTAN$;TAB(11);DS$;
790 VT=3:GOSUB930
800 T=0
810 FORX=1TO17
820 IFX>NTHENLPRINT:GOTO890
830 LPRINTUSING"#####";Q(X);
840 LPRINTTAB(20);D$(X);
850 LPRINTTAB(63);:LPRINT USING"####.##"
    ;UP(X);
860 ST=Q(X)*UP(X)
870 LPRINTTAB(72);
880 LPRINTUSING"#####.##";ST:T=T+ST
890 NEXT
900 LPRINTTAB(60);" ";TAB(11)" ";
910 LPRINTUSING"#####.##";T
920 GOTO980
930 REM VERTICAL TAB HERE
940 FOR V=1 TO VT
950 LPRINT
960 NEXT
970 RETURN
980 END

```


Program Listing for STATEM

```
10 REM 1620 bytes
20 CLS
30 CLEAR350
40 DATA"Your Company Name"
50 DATA"Your Street Address"
60 DATA"Town, State, and Zip"
70 DATA"Your Phone Number"
80 DIM C$(3),N$(3),DT$(14),CR$(14),A(14)
90 L=0
100 FORX=1TO3
110 READ C$(X):IFLEN(C$(X))>LTHENL=LEN(C
    $(X))
120 NEXT
130 READPH$
140 FORX=1TO3:IFLEN(C$(X))=LTHEN170
150 DI=(L-LEN(C$(X)))/2
160 C$(X)=STRING$(DI,32)+C$(X)
170 NEXT
180 PRINT"Statement Form for:":PRINT
190 FORX=1TO3:PRINTTAB(10);C$(X)
200 NEXT
210 PRINT"Company statement is going to?"
    "
220 LINEINPUTN$(1)
230 PRINT"Street address?"
240 LINEINPUTN$(2)
250 PRINT"Town, State, and Zip code?"
260 LINEINPUTN$(3)
270 L=0:FORX=1TO3
280 IFLEN(N$(X))>LTHENL=LEN(N$(X))
290 NEXT
300 FORX=1TO3
310 IFLEN(N$(X))=LTHEN340
320 DI=(L-LEN(N$(X)))/2
330 N$(X)=STRING$(DI,32)+N$(X)
340 NEXT
350 CLS:PRINT"Statement going to:":PRINT
360 FORX=1TO3:PRINTTAB(10);N$(X):NEXT
370 INPUT"Date for statement";D$
380 INPUT"Account number";AN$
390 CLS:PRINT"Input charges and credits
    (14 maximum)."
```

400 PRINT"Null to end input."

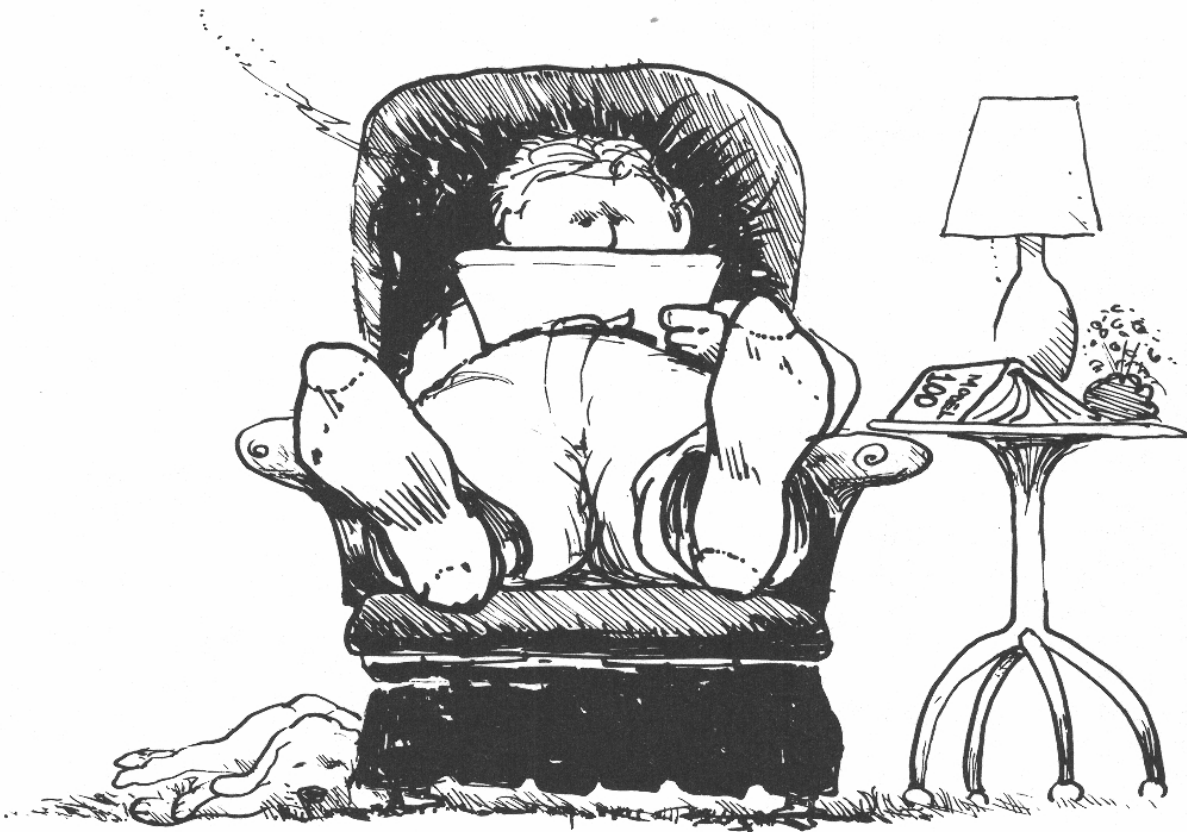
410 N=1

```

420 PRINT"Charge #";N;"?";
430 LINEINPUTCR$(N)
440 IFCR$(N)=" "THENN=N-1:GOTO490
450 INPUT"Date";DT$(N)
460 INPUT"Amount";A(N)
470 N=N+1:IFN>14THEN490
480 GOTO420
490 CLS:T=0:PRINT"Ready Printer,"
500 PRINT"Then press ENTER."
510 A$=INKEY$:IFA$<>CHR$(13)THEN510
520 CLS
530 PRINT"Does your form have your compa
    ny"
540 PRINT"name on it (Y/N)?"
550 A$=INKEY$:IFA$=" "THEN550
560 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
570 IFA$="Y"THEN690
580 IFA$<>"N"THEN550
590 VT=3:GOSUB870
600 LPRINTTAB(12);C$(1)
610 LPRINTTAB(12);C$(2);TAB(52);D$
620 LPRINTTAB(12);C$(3)
630 LPRINTTAB(52);AN$
640 VT=2:GOSUB870
650 VT=2:GOSUB870
660 LPRINTTAB(22);PH$
670 VT=2:GOSUB870
680 GOTO730
690 VT=6:GOSUB590
700 LPRINTTAB(52);D$
710 LPRINTTAB(52);AN$
720 VT=5:GOSUB870
730 FORX=1TO3
740 LPRINTTAB(12);N$(X)
750 NEXT
760 VT=8:GOSUB870
770 T=0
780 FORX=1TON
790 LPRINTTAB(6);DT$(X);
800 LPRINTTAB(22);CR$(X);
810 LPRINTTAB(62);
820 LPRINTUSING"#####.##";A(X)
830 T=T+A(X)
840 NEXT
850 LPRINT:LPRINTTAB(50);"TOTAL";TAB(62)
    ;T
860 END
870 FORV=1TOVT

```

880 LPRINT
890 NEXT:RETURN



Ω

PART III

Home Programs

This section is devoted to programs for use at home, although several of them can also be used for businesses.

A perpetual calendar is the first program in this section. It'll tell you the day of the week of any date (for example, that January 19, 1983, my birthday, was a Wednesday), display a calendar for any month, or tell you the number of days between any two dates (useful if you want to calculate interest on accounts deposited for specific time periods).

LOANS1 is a program to help you determine the interest rate of a loan.

If you have to divide each month's phone bill between two or more people or businesses (as I separate my business calls from personal), then PHONE is the program you need.

Similarly, if you have trouble reconciling your checkbook to the bank statement, try CHECK; it might solve your problems.

Balancing a budget is always a big problem, especially trying to decide how much to pay on those bills while still trying to keep some extra spending money. BUDGET will make it a bit easier to do that chore.

Keeping track of your car's miles per gallon can alert you to problems long before something breaks and you have to haul the hulk in for repairs. Simplifying that problem is MILE.

And if you're trying to figure out if it's time to get a new car, maybe you should first figure out what it costs you to keep your current vehicle in operation. Or if you just bought a new car, maybe you should keep track of those expenses to see just how much money you spend on it. The answer may surprise you, or even change some of your driving habits. In any case, try out the MAIN program.

When I order something through the mail, I use the MAIL program to list everything I ordered clearly and legibly, as well as giving my address and totaling the order for me.



CALNDR

Create a perpetual calendar on your printer.

Calendar is a simple program that calculates the number of days between any two dates, or will tell the day of the week that any given date falls on, or it will give you a monthly calendar for any month of any year, including leap years.

Once you have the calendar for the month you want, use the print button on your Model 100 to produce a fast and convenient printout.

If you want to try a little programming exercise, change line 290 to a FOR...NEXT loop that counts from 1 to 12, change the PRINT commands in lines 320, 340, 380 and 390 to LPRINTs, and change line 410 to "LPRINT:LPRINT:NEXT:RETURN." Hook up a printer and you have a program that prints out an entire year's calendar for you.

Program Listing for CALNDR

```
5 REM 1429 bytes
10 DATASUNDAY,MONDAY,TUESDAY,WEDNESDAY,T
   HURSDAY,FRIDAY,SATURDAY
20 DATA31,28,31,30,31,30,31,31,30,31,30,
   31
30 CLEAR200:MAXFILES=0
40 DIMD$(6),DM(12)
50 FORX=0TO6
60 READD$(X)
70 NEXT
80 FORX=1TO12
90 READDM(X)
100 NEXT
110 CLS
```

```

120 PRINTTAB(16)"Calendar"
130 PRINTTAB(8);"1 Look for one day."
140 PRINTTAB(8);"2 Print Calendar for on
    e month."
150 PRINTTAB(8);"3 For days between date
    s."
160 PRINT:INPUT"Your selection";I
170 IF I<1 OR I>3 THEN 110
180 CLS
190 IF I=2 THEN 280
200 IF I=3 THEN 420
210 INPUT"YEAR";YY:IF YY<0 THEN 210
220 INPUT"MONTH";MM:IF MM>12 OR MM<1 THEN 220
230 INPUT"DAY";DD:IF DD<1 OR DD>31 THEN 230
240 GOSUB 600
250 GOSUB 650
260 PRINT"THAT DAY IS ";D$(DW)
270 GOTO 550
280 INPUT"YEAR";YY:IF YY<0 THEN 280
290 INPUT"MONTH";MM:IF MM<1 OR MM>12 THEN 290
300 CLS
310 FOR X=0 TO 6
320 PRINTTAB((X*5)+5);LEFT$(D$(X),3);
330 NEXT
340 PRINT
350 FOR DD=1 TO DM(MM)
360 GOSUB 600
370 GOSUB 650
380 PRINTTAB((DW*5)+5);DD;
390 IF DW=6 THEN PRINT
400 NEXT
410 GOTO 550
420 CLS:PRINT"DAYS BETWEEN DATES"
430 INPUT"FIRST YEAR";YY:IF YY<0 THEN 430
440 INPUT"FIRST MONTH";MM:IF MM<1 OR MM>12 THEN 440
450 INPUT"FIRST DAY";DD:IF DD<1 OR DD>31 THEN 450
460 GOSUB 600
470 F1=FA
480 CLS:INPUT"SECOND YEAR";YY:IF YY<0 THEN 480
490 INPUT"SECOND MONTH";MM:IF MM<1 OR MM>12 THEN 490
500 INPUT"SECOND DAY";DD:IF DD<1 OR DD>31 THEN 500
510 GOSUB 600
520 F2=FA

```

```

530 DS=ABS(F1-F2)
540 PRINTDS;"DAYS"
550 PRINT@280,"AGAIN Y/N?";
560 A$=INKEY$:IFA$=""THEN560
570 IFA$="Y"ORA$="y"THEN30
580 IFA$<>"N"ANDA$<>"n"THEN560
590 END
600 IFMM=1ORMM=2THEN630
610 FA=365*YY+DD+31*(MM-1)-INT(.4*MM+2.3
    )+INT(YY/4)-INT(3/4*(INT(YY/100)+1)
    )
620 RETURN
630 FA=365*YY+DD+31*(MM-1)+INT((YY-1)/4)
    -INT(3/4*(INT((YY-1)/100)+1))
640 RETURN
650 REM----- DAY OF WEEK -----
660 DW=FA+(-INT(FA/7)*7)
670 DW=DW-1
680 IFDW<0THENDW=6
690 RETURN

```

Ω

LOANS1

Determine the interest rate on a loan.

This program will calculate the interest rate of a loan when given the present value of the loan, its duration, the number of compounding periods per year, and the amount of the payments.

Once the interest rate is calculated, you can change the values of the variables.

This program is handy when you want to borrow money and want to see the different interest rates for different payments. Selecting the payment you'd like with the amount you want to borrow, you may find the interest rate is too low for a bank to bother with. In which case you'll have to adjust your budget to accommodate the interest rate your bank will probably want. (As I'm typing this the interest rate on a \$1000 signature loan is 17%, with either one or two years payback time; I get to choose which one I want.)

Program Listing for LOANS1

```
5 REM 986 bytes
10 CLEAR256:CLS:MAXFILES=0:DEFDBLI
20 PRINTTAB(12)"Loan Interest Rate"
30 INPUT"Present Value (dollars)";PV
40 IFA=1THEN120
50 INPUT"Number of periods per year";NY
60 IFA=3THEN120
70 INPUT"Duration of loan (periods)";N
80 IFA=2THEN120
90 INPUT"Amount of each payment (dollars
   )";P
100 IFA=4THEN120
110 PRINT"One moment please, I'm calcula
```

```

      ting."
120 I=.008
130 I1=P/PV*((1+I)^N-1)/(1+I)^N
140 IFABS(I-I1)<.000001THEN170
150 I=I1
160 GOTO130
170 I=I1*NY*100
180 CLS:PRINT
190 PRINTUSING"Present Value          $$##
      #,###.##";PV
200 PRINTUSING"Number of periods per yea
      r      ###";NY
210 PRINTUSING"Duration of loan (periods
      )      ###";N
220 PRINTUSING"Amount of each payment $$
      #,###.##";P
230 PRINTUSING"Interest Rate is
      ###%";I
240 PRINT@287,"Press <ENTER> to continue
      ";
250 IFINKEY$<>CHR$(13)THEN250
260 A=0:CLS:PRINT"DO YOU WANT TO:"
270 PRINT"      1 - Change loan value"
280 PRINT"      2 - Change num. of interest
      periods"
290 PRINT"      3 - Change periods per year
      "
300 PRINT"      4 - Change payment amount"
310 PRINT"      5 - Re-enter information"
320 PRINTTAB(13);:INPUT"Your Choice";A
330 IFA<1ORA>5THEN320
340 ONAGOTO30,70,50,90,10

```

PHONE

Separate and tally business calls on your home telephone.

This program was developed to help me figure out how much of my phone bill I could write off to business expense. Since I'm an editor for a magazine, in addition to being a freelance consultant, I have to separate my business calls from calls made for the magazine, meaning three categories.

To use the program, type in DATA statements for the different categories for your phone bill (lines 30-60). The maximum number is eight. When you use the program, the categories are listed on the right side of the display. On the left side you're prompted for the cost of the first long distance phone call, then asked which category it belongs in. As you enter each phone call and its category, running totals are kept by the program. When you've reached the end of the list of calls, enter a zero for the call's cost.

The screen is cleared and the subtotal for each category is displayed along with their total. Compare this total with the total for the long distance calls in your phone statement. They should match.

Next you're asked for the monthly service charge for the phone and equipment on your statement, and the federal tax on the long distance calls.

With all this information, the service charge and federal taxes are divided up by percentages between the different categories, and the totals displayed for you to record: the category, its long distance call subtotal, its proportionate service charge and tax, and its final total. Since the display of the categories erases the previous screen's display, you can have the program repeat the previous display.

Program Listing for PHONE

5 REM 1163 bytes

```

10 CLS: CLEAR 256: PRINT "Phone Bill Calculator"
20 REM ENTER CATEGORIES HERE (8 MAX.)
30 DATA Personal
40 DATA Business
50 DATA M100
60 DATA 00
70 DIM C$(8), P(8): MAXFILES = 0
80 N = 1
90 READ C$(N)
100 IF C$(N) = "00" THEN N = N - 1: GOTO 130
110 PRINT @23 + (N - 1) * 40, USING "# \
    \"; N; C$(N);
120 IF N < 8 THEN N = N + 1: GOTO 90
130 PRINT @80, "Cost for call, ": INPUT "0 to
    end"; CC
140 IF CC = 0 THEN 210
150 PRINT @160, " ";
160 INPUT "What category"; A$
170 C = VAL(A$): IF C < 1 OR C > 8 THEN 150
180 T(C) = T(C) + CC: T = T + CC
190 PRINT @120, STRING$(22, 32);: PRINT @160,
    STRING$(22, 32);
200 PRINT @280, T;: GOTO 130
210 CLS
220 TC = 0: FOR X = 1 TO N: TC = TC + T(X): NEXT
230 PRINT USING "Total for calls = $$$###, #
    ##.##"; TC
240 FOR X = 1 TO N
250 P(X) = (T(X) * 100) / TC
260 PRINT USING "\
    \ = ##.##% "; C$(
    X); P(X),
270 NEXT: PRINT @240, " ";
280 INPUT "Monthly Service Charge"; MC
290 IF MC <= 0 THEN 280
300 INPUT "Federal Tax this month"; FT
310 IF FT <= 0 THEN 300
320 GOSUB 430: TP = 0: CP = 0: FOR X = 1 TO N
330 IF X / 6 = INT(X / 6) THEN GOSUB 410
340 PT = (P(X) / 100) * FT: TP = TP + PT
350 PC = (P(X) / 100) * MC: CP = CP + PC
360 PRINT USING "\
    \ $$$###.## $$$###.##
    $$$###.## $$$###.##"; C$(X); T(X); PC; PT; T
    (X) + PC + PT;
370 NEXT
380 PRINT @240, USING " TOTALS $$$###.## $$$
    ### $$$###.## $$$###.##"; TC; CP; TP; TC + C
    P + TP;

```

```
390 PRINT@280,"Repeat list";:INPUTA$
400 IFA$="y"ORA$="Y"THEN210ELSEEND
410 PRINT@284,"Press <ENTER> to continue
    ";
420 IFINKEY$<>CHR$(13)THEN420
430 CLS:PRINT"Category   Calls   Ser/Chg
      Tax   Total":RETURN
```

Ω



CHECK

Reconcile your checkbook the easy way.

Balancing your checkbook is one of life's more arduous tasks. I used to be able to get my checkbook to balance every time, but I seem to have lost that talent several years ago. Hence, this program.

When your next bank statement arrives, run this program and type in the number of the last check you wrote. Then type in any deductions you may have from your checking account that aren't listed on your bank statement (such as withdrawals from an automatic teller machine, or bills that are automatically deducted from your account). Next type in any deposits that aren't listed on the bank statement, and the ending balance shown on your statement.

Now the program is going to step through the checks, from the low number you gave it to the last check you wrote, asking for the amount of each one. Voided checks are indicated by a zero. Just copy these from your checkbook.

Now you go through the list a second time, indicating which checks have cleared the bank. Then, as a review, any checks you haven't cleared will be listed, one at a time, and you'll be asked for a Yes/No response on whether or not it cleared.

The checks that haven't cleared will be listed, and a total for the outstanding checks will be displayed. Other deductions will be listed, as will deposits not on the bank statement. Finally, the balance that should be in your checkbook will be listed.

If they're different, make sure that you've subtracted all bank charges from your checkbook balance. If they still disagree, have someone else try it. Maybe you forgot something, or accidentally included something.

If you want to practice your programming skills, you could add the ability to save uncleared checks to tape, so you won't have to retype them next month. To do this you'll need a FOR...NEXT loop to remove the cleared checks and move

the others to the beginning of the array, and a quick list feature so you'll know what the numbers are when you load them from tape.

Program Listing for CHECK

```
10 REM 1200 bytes
20 MAXFILES=0: CLEAR256: CLS
30 DIMCN(100), CF(100), CA(100), CC(100)
40 PRINTTAB(10) "Check Book Balancing"
50 INPUT "Input lowest check #"; LC
60 INPUT "Input highest check #"; HC
70 PRINT "Total of other deductions not l
   isted on"
80 INPUT "your statement"; OD
90 PRINT: INPUT "Total Deposits not on sta
   tement"; DP
100 INPUT "Statement ending balance" BA
110 CLS
120 PRINT "Enter a zero for void checks."
130 C=1
140 PRINT: FORX=LCTOHC
150 PRINT "Check #"; X;
160 INPUT "   Amount="; CA(C)
170 CN(C)=X
180 C=C+1
190 NEXT
200 TC=C: C=1
210 REM   TC= TOTAL NUMBER OF CHECKS TO
   BALANCE
220 REM
230 CLS: PRINT "Cleared check input starts
   now."
240 PRINT "Enter a zero for check number
   when done."
250 PRINT "Check #"; C;
260 INPUT CC(C)
270 IF CC(C)=0 THEN 300
280 C=C+1
290 GOTO 250
300 CLS
310 FORX=1 TO C
320 FORN=1 TO TC
330 IF CN(N)=CC(X) THEN CF(N)=1
340 NEXT
350 NEXT
360 CLS
370 PRINT "Checks not cleared yet.": PRINT
```

```

380 FORX=1TOTC
390 IFCF(X)<>1THENPRINT"Check #";CN(X);T
    AB(15);"$";CA(X)
400 IFCF(X)=1THEN440
410 PRINT"Cleared (Y/N)?"
420 A$=INKEY$:IFA$=""THEN420
430 IFA$="Y"ORA$="y"THENCN(X)=1
440 IFCF(X)<>1THENT=T+CA(X)
450 NEXT
460 CLS:T=0
470 PRINT"Checks not cleared yet":PRINT
480 FORX=1TOTC
490 IFCF(X)<>1THENPRINT"Check #";CN(X);T
    AB(15);"$";CA(X)
500 IFCF(X)<>1THENT=T+CA(X)
510 NEXT
520 PRINTTAB(10);"Total";TAB(15);T
530 PRINT"Other deductions =";OD
540 B=BA-T-OD
550 PRINT"Deposits";DP:B=B+DP
560 PRINT"Your balance is";B

```



BUDGET

Set up and track your household budget.

Everyone needs a budget, and this program will help you establish one. The biggest problem with a budget is deciding how much to pay whom in such a way as to meet all your obligations while still having a little cash left over for extras. BUDGET will let you arrange a budget, then change amounts for the different items until you reach a satisfactory setup.

First, type in the different categories you want in your budget (there's an arbitrary limit of thirty, which can be changed by increasing or decreasing the dimension size of the variables in line 30). When you've entered them all, press ENTER in response to the "Category?" prompt. Next, assign a dollar figure to each category, and a number for your income. If you're satisfied with the result, press ENTER to end the program otherwise press "C" to change a category's value. Select the category's number, then enter the new dollar amount beside the old one. The budget list will be redisplayed and the money left over refigured.

Program Listing for BUDGET

```
10 REM 790 bytes
20 MAXFILES=0: CLEAR300:CLS
30 DIMC$(30),S(30):PRINTTAB(13)"Budget "
40 N=1
50 PRINT"Enter null to end input."
60 PRINT"Category #";N;
70 INPUTC$(N)
80 IFC$(N)="" THENN=N-1:GOTO100
90 N=N+1:GOTO60
100 CLS
```

```

110 TN=N
120 N=1
130 PRINT"How much to spend on ";C$(N);
140 INPUTS(N)
150 N=N+1:IFN>TNTHEN170
160 GOTO130
170 CLS:INPUT"Your income";IN
180 CLS
190 T=0
200 FORX=1TOTN:IFX/13=INT(X/13)THENGOSUB
    350
210 PRINTUSING"\          \#####.## ";C$(
    X);S(X);
220 T=T+S(X)
230 NEXT:PRINT
240 PRINTUSING"TOTAL $$#####.## left ove
    r: $$#####.##";T;IN-T
250 PRINT@284,"Press ENTER to end, C to
    change.";
260 A$=INKEY$:IFA$=""THEN260
270 IFA$="C"ORA$="c"THEN290
280 END
290 CLS:FORX=1TOTN:IFX/13=INT(X/13)THENG
    OSUB350
300 PRINTUSING"## \          \#####.## ";X;C
    $(X);S(X);:NEXT
310 PRINT:INPUT"Which category #";CN
320 PRINTUSING"Old amount$$#####.## ";S(C
    N);
330 INPUT"New amount";S(CN)
340 GOTO180
350 PRINT@287,"Press ENTER to continue";
360 IFINKEY$<>CHR$(13)THEN360ELSECLS:RET
    URN

```

Ω

MILE

Track mileage for your auto.

One way to keep track of your car's performance and catch troubles before they become a real problem is to watch the miles per gallon your car gets. If the mileage begins to drop, something is wrong. Perhaps the car needs a tuneup, or an oil change, or new gas or air filter.

Watching your mileage will also show you how driving habits can eat up gas faster than normal. With most cars, driving in excess of 50 miles per hour will result in higher gas consumption. Some cars, however, get better mileage at 65 than 55, especially the diesel fueled vehicles.

This program lets you type in the number of gallons your car took, the current odometer reading, and the cost of the fuel. If you happen to fill up at a station that gives readings in liters instead of gallons, no problem. Just add an "L" to the end of the amount of fuel and the program will convert it to gallons.

This program is arbitrarily limited to 50 lines of data. This can be changed by altering the dimension statement.

When you're done entering information, you can save the data to memory by specifying a six-character filename. If you want, you can save the data to other devices by specifying the appropriate device prefix. For example, to save the file TEST to tape, you'd use the filename "CAS:TEST," to save it out the modem line, type "M7E1E:TEST," where M stands for modem, seven for seven data bits per character, E for Even parity checking, 1 for one stop bit after the character, and the last E is for Exon/Exoff protocol.

To load data, use the same filename and prefix used when you saved the data.

After you have your data in the program, you can list it to the display. Your mileage is computed based on the number of miles you've driven between your last fuel stop and the next one. This mileage will be accurate only if you always fill the tank. Not filling the tank will give you an inaccurate figure, either

artificially low or high.

The final display will give you the total mileage you've driven, your accumulated miles per gallon (which will be very accurate if you haven't forgotten to include any fuel stops, and doesn't include the fuel added at the last stop since that fuel hasn't been driven off yet), and your car's gas cost in dollars per mile.

This program was designed to interact with data produced by the next program, "MAIN."

Program Listing for MILE

```
10 REM 1970 bytes
20 MAXFILES=1: CLEAR1000: DIMM(2,50),M$(50)
30 CLS:PRINTTAB(6)"Car Mileage/Costs"
40 PRINT"1. Mileage"
50 PRINT"2. List data"
60 PRINT"3. Read data"
70 PRINT"4. Write data"
80 PRINTTAB(13)"Your choice";
90 INPUTA$:IFA$=""THEN90ELSEA=VAL(A$)
100 IFA<1ORA>4THEN20
110 CLS:ONAGOTO550,220
120 INPUT"Filename";F$
130 IFF$=""THEN120
140 IFA<>3THEN190
150 OPENF$FORINPUTAS1:F=0:PRINT"Reading
    data"
160 IFEOF(1)THENCLOSE:GOTO180
170 F=F+1:INPUT#1,M$(F),M(0,F),M(1,F),M(
    2,F):GOTO160
180 GOTO30
190 OPENF$FOROUTPUTAS1:PRINT"Writing dat
    a"
200 FORI=1TOF:PRINT#1,M$(I):PRINT#1,M(0,
    I);M(1,I);M(2,I):NEXT
210 CLOSE:GOTO30
220 INPUT"List to display or printer (D/
    P)";A$
230 IFA$<>"P"ANDAS$<>"p"THEN370
240 LPRINT"Car Mileage":GOSUB540
250 G=0:G1=0:R=0:M=0:FORI=1TOF
260 IFRIGHT$(M$(I),1)<>"A"THEN320
270 LPRINTLEFT$(M$(I),2)"/"MID$(M$(I),3,
    2)"/"MID$(M$(I),5,2);
280 LPRINTTAB(10)M(0,I);TAB(20)M(1,I);TA
    B(30)"$M(2,I);TAB(40);
290 IFI>1THENLPRINT(M(0,I)-M(0,I-1))/M(1
```

```

      ,I)ELSELPRINT
300 G=G+M(1,I):G1=G1+M(2,I)
310 IF I/60=INT(I/60) THEN LPRINT CHR$(12):G
      OSUB540
320 NEXT:LPRINT"Total mileage ="M(0,F)-M
      (0,1)
330 LPRINT"Total gallons ="G
340 LPRINT"Fuel cost = $"G1
350 LPRINT"Miles per gallon ="(M(0,F)-M(
      0,1))/G
360 LPRINT"Cost per mile ="G/(M(0,F)-M(
      0,1)):GOTO30
370 GOSUB530:G=0:G1=0:FOR I=1 TO F
380 IF RIGHT$(M$(I),1) <> "A" THEN 440
390 PRINT LEFT$(M$(I),2) "/" MID$(M$(I),3,2)
      "/" MID$(M$(I),5,2);
400 PRINT TAB(8)M(0,I);TAB(18)M(1,I);TAB(
      26)"$M(2,I);TAB(33);
410 IF I>1 THEN PRINT USING"###.#";(M(0,I)-M
      (0,I-1))/M(1,I) ELSE PRINT
420 G=G+M(1,I):G1=G1+M(2,I)
430 IF INT(I/6)=I/6 THEN GOSUB510
440 NEXT:GOSUB520:CLS
450 PRINT"Total mileage ="M(0,F)-M(0,1)
460 PRINT USING"Total gals. = #####.#";G
470 PRINT USING"Fuel cost = $$###,###.##";G
      1
480 PRINT"Miles per gallon ="(M(0,F)-M(0
      ,1))/(G-M(1,F))
490 PRINT"Cost per mile = $"G1/(M(0,F)-M
      (0,1))
500 GOSUB510:GOTO30
510 PRINT@287,"Press ENTER to continue."
      ;
520 IF INKEY$ <> CHR$(13) THEN 520
530 CLS:PRINT"   Date"TAB(10)"Mileage"TAB
      (18)"Gallons"TAB(26)"Cost"TAB(35)"M
      PG":RETURN
540 LPRINT:LPRINT"Date"TAB(10)"Mileage"TAB
      (20)"Gallons"TAB(30)"Cost"TAB(40)
      "MPG":RETURN
550 INPUT"Date (MM/DD/YY format)";D$
560 IF D$="" THEN 30
570 IF D$="" THEN 550
580 IF LEN(D$) <> 8 THEN 550
590 D$=LEFT$(D$,2)+MID$(D$,4,2)+RIGHT$(D
      $,2)+"A"
600 INPUT"Mileage";M:IF M<1 THEN 600

```



```

610 G=0:IFA<>1THEN650
620 INPUT"Gallons";G$:IFG$=""THEN620
630 G=VAL(G$):IFG=0THEN620
640 IFRIGHT$(G$,1)="1"ORRIGHT$(G$,1)="L"
    THENG=G/3.84615
650 INPUT"Cost";C:IFC<1THEN650
660 F=F+1:M$(F)=D$:M(0,F)=M:M(1,F)=G:M(2
    ,F)=C
670 GOTO550

```



Ω

MAIN

An automobile maintenance program.

How much does your car cost you to run? What does it cost you on a per mile basis? On a per mile basis, would it be cheaper to buy a new car with increased gas mileage and lower maintenance costs?

If you're curious about questions like these, this program will help answer them for you.

Like the previous program, MILE, you type in your car's odometer reading whenever you have gas, maintenance, or repair work done to your car. In addition, you type in the cost of these services. If you buy gas, you also type in the number of gallons, or liters (appending an "L" to the end of the number), you put in.

When you've finished entering data you can save the information to memory or tape (see MILE for details), display it, or print it out on a printer.

Saving and loading data is the same for both MILE and MAIN. In fact, you can both read and write files created by either program with either program. Thus, if you have a data file created by MAIN, and you want to know how many miles per gallon your car gets, just use the MILE program to read in the data file and compute the mileage. Similarly, you can take a MILE data file and make it into a MAIN data file by loading it in and adding data for repairs and maintenance costs.

When you're ready, the program will run through the data and compute your car's total repair costs, total maintenance costs, total fuel costs, and the bottom line cost to run your car, as a total and on a per mile basis.

With this information, you can decide whether or not your current car is worth the money you spend on it. You can also figure out how much money you have to budget for your car based on the miles you drive it, or intend to drive it. (Such as on an extended vacation: if you know your car's per mile expenses, and

the total mileage of the vacation, give or take a hundred, you can decide how much money to take along to keep the car in running order.)

Like MILE, this program is arbitrarily set to a limit of fifty items in its list. Changing the dimension statement will fix that if fifty is too small a number.

Program Listing for MAIN

```
10 REM 2311 bytes
20 MAXFILES=1: CLEAR1000: DIMM$(50), M(2,50)
30 CLS: PRINTTAB(6) "Car Mileage/Maintenan
    ce Costs"
40 PRINT "1. Mileage"
50 PRINT "2. Maintenance"
60 PRINT "3. Repairs"
70 PRINT "4. List data"
80 PRINT "5. Read data"
90 PRINT "6. Write data"
100 PRINTTAB(13) "Your choice";
110 INPUT A$: IFA$ = "" THEN 110 ELSE A$ = VAL(A$)
120 IFA < 1 OR A > 6 THEN 20
130 CLS: IFA < 4 THEN 600
140 IFA < 5 THEN 250
150 INPUT "Filename"; F$
160 IFF$ = "" THEN 150
170 IFA < > 5 THEN 220
180 OPEN F$ FOR INPUT AS 1: F = 0: PRINT "Reading
    data"
190 IF EOF(1) THEN CLOSE: GOTO 210
200 F = F + 1: INPUT #1, M$(F), M(0,F), M(1,F), M(
    2,F): GOTO 190
210 GOTO 30
220 OPEN F$ FOR OUTPUT AS 1: PRINT "Writing dat
    a"
230 FOR I = 1 TO F: PRINT #1, M$(I): PRINT #1, M(0,
    I); M(1,I); M(2,I): NEXT
240 CLOSE: GOTO 30
250 INPUT "List to display or printer (D/
    P)"; A$
260 IFA$ < > "P" AND A$ < > "p" THEN 400
270 LPRINT "Car Mileage/Maintenance Costs
    ": GOSUB 590
280 G = 0: G1 = 0: R = 0: M = 0: FOR I = 1 TO F
290 LPRINT LEFT$(M$(I), 2) "/" MID$(M$(I), 3,
    2) "/" MID$(M$(I), 5, 2);
300 LPRINTTAB(10) M(0,I); TAB(20) M(1,I); TA
    B(30) "$" M(2,I); TAB(40);
310 IF RIGHT$(M$(I), 1) = "A" THEN LPRINT "Gas "
```

```

      :G=G+M(1,I):G1=G1+M(2,I)
320 IFRIGHT$(M$(I),1)="B"THENLPRINT"Main
    tenance":M=M+M(2,I)
330 IFRIGHT$(M$(I),1)="C"THENLPRINT"Repa
    ir":R=R+M(2,I)
340 IFI/60=INT(I/60)THENLPRINTCHR$(12):G
    OSUB590
350 NEXT:LPRINT"Total mileage ="M(0,F)-M
    (0,1)
360 LPRINT"Total gallons ="G;TAB(30)"Fue
    l cost = $"G1
370 LPRINT"Miles per gallon ="(M(0,F)-M(
    0,1))/G
380 LPRINT"Repairs cost = $"R;TAB(30)"Ma
    intenance cost = $"M
390 LPRINT"Total operating cost = $"G1+R
    +M;TAB(30)"or $"(M+G1+R)/(M(0,F)-M(
    0,1))"per mile.":GOTO30
400 GOSUB580:G=0:G1=0:R=0:M=0:FORI=1TOF
410 PRINTLEFT$(M$(I),2)"/"MID$(M$(I),3,2
    )"/"MID$(M$(I),5,2);
420 PRINTTAB(8)M(0,I);TAB(18)M(1,I);TAB(
    26)"$M(2,I);TAB(33);
430 IFRIGHT$(M$(I),1)="A"THENG=G+M(1,I):
    G1=G1+M(2,I):PRINT"Gas"
440 IFRIGHT$(M$(I),1)="B"THENPRINT"Main.
    ":M=M+M(2,I)
450 IFRIGHT$(M$(I),1)="C"THENPRINT"Repai
    r":R=R+M(2,I)
460 IFINT(I/6)=I/6THENGOSUB560
470 NEXT:GOSUB570:CLS
480 PRINT"Total mileage ="M(0,F)-M(0,1)
490 PRINTUSING"Total gals.=#####.# Fuel
    cost=$#####.##";G;G1;
500 PRINT"Miles per gallon ="(M(0,F)-M(0
    ,1))/(G-M(1,F))
510 PRINT"Repairs cost = $"R
520 PRINT"Maintenance cost = $"M
530 PRINT"Total operating cost = $"G1+R+
    M
540 PRINT"or $"(M+G1+R)/(M(0,F)-M(0,1))"
    per mile."
550 GOSUB560:GOTO30
560 PRINT@287,"Press ENTER to continue."
    ;
570 IFINKEY$<>CHR$(13)THEN570
580 CLS:PRINT" Date"TAB(10)"Mileage"TAB
    (18)"Gallons"TAB(26)"Cost":RETURN

```

```

590 LPRINT:LPRINT"Date"TAB(10)"Mileage"TAB(20)"Gallons"TAB(30)"Cost":RETURN

600 INPUT"Date (MM/DD/YY format)";D$
610 IFD$=""THEN30
620 IFD$=""THEN600
630 IFLEN(D$)<>8THEN600
640 D$=LEFT$(D$,2)+MID$(D$,4,2)+RIGHT$(D$,2)+CHR$(64+A)
650 INPUT"Mileage";M:IFM<1THEN650
660 G=0:IFA<>1THEN700
670 INPUT"Gallons";G$:IFG$=""THEN670
680 G=VAL(G$):IFG=0THEN670
690 IFRIGHT$(G$,1)="1"ORRIGHT$(G$,1)="L"
    THENG=G/3.84615
700 INPUT"Cost";C:IFC<1THEN700
710 F=F+1:M$(F)=D$:M(0,F)=M:M(1,F)=G:M(2,F)=C
720 GOTO600

```



Ω

MAIL

A computerized order form.

When you order things by mail, there's always the question of whether or not the receiver of your order can read your writing. One way around this problem is to use a typewriter. While it does make your order more legible, you also tend to make slight typing errors. Plus, moving the typing carriage around to make everything fit right on the paper is a hassle.

This program takes the catalog numbers, quantities, descriptions and costs of the items you want, and generates an order form on your printer. It prints your name, address and phone number, as well as a short message advising the recipient of your order, lists the items in columns, and prints a total for the order.

If there's a postage and handling charge, you can easily modify the program to handle that.

Program Listing for MAIL

```
10 REM 1124 bytes
20 CLEAR350
30 CLS
40 DATA"Your name"
50 DATA"Your street address"
60 DATA"Your town, state, and zip code"
70 DATA"Your phone number"
80 DIM C$(3),I$(20),Q(20),C(20)
90 L=0
100 FORX=1TO3
110 READ C$(X):IFLEN(C$(X))>LTHENL=LEN(C
    $(X))
120 NEXT
```

```

130 READPH$
140 FORX=1TO3:IFLEN(C$(X))=LTHEN170
150 DI=(L-LEN(C$(X)))/2
160 C$(X)=STRING$(DI,32)+C$(X)
170 NEXT
180 PRINT"Enter the items you wish to or
    der."
190 PRINT"(20 maximum, Null to end input
    .)"
200 N=1
210 PRINT"Description of item"N"?";:LINE
    INPUTI$(N)
220 IFI$(N)=""THENN=N+1:GOTO260
230 INPUT"Quantity required?";Q(N)
240 INPUT"Cost";C(N)
250 N=N+1:GOTO210
260 CLS:PRINT"Ready your printer and pre
    ss ENTER."
270 A$=INKEY$:IFA$<>CHR$(13)THEN270
280 VT=3:GOSUB500
290 FORX=1TO3:LPRINT TAB(10);C$(X):NEXT
300 VT=3:GOSUB500
310 LPRINT"Dear sirs,"
320 LPRINT"    Listed below are the item
    s I"
330 LPRINT"wish to order from your compa
    ny."
340 LPRINT"Enclosed with this order is a
    check for the total amount of the
    order."
350 VT=2:GOSUB500
360 LPRINT"QTY";TAB(15);"DES.";TAB(32);"
    COST";TAB(42);"EXT COST"
370 T=0
380 FORX=1TON
390 LPRINTUSING"####";Q(X);
400 LPRINTTAB(15);I$(X);
410 LPRINTTAB(30);
420 LPRINTUSING"#####.##";C(X);
430 T=T+(C(X)*Q(X))
440 LPRINTTAB(40);
450 LPRINTUSING"#####.##";C(X)*Q(X)
460 NEXT
470 LPRINTTAB(30);"TOTAL";TAB(40);
480 LPRINTUSING"#####.##";T
490 END
500 FORV=1TOVT:LPRINT:NEXT:RETURN

```

PART IV

Games

No general interest book on computer programs would be complete without a few games; therefore, we have here, games!

Most of these programs will fit in an 8K machine with plenty of room left over for other programs, but some of them are rather large and may require a 16K RAM machine. We wrote these games in a fairly straightforward manner, keeping most instructions on separate lines. Some lines have more than one instruction, but most don't.

If you're pressured for room, you can reduce these programs by removing the REM statements from them.

Another space-saving trick is to use the edit command to combine several instructions onto one line. A word of caution is needed about doing that: Make sure that the lines you combine aren't addressed by a GOTO, IF...THEN or GOSUB statement. That is; if you combine lines 10, 20, 30, 40 and 50 into line 10, you'll have problems if somewhere else in your program there's a GOTO30 statement.

Another programming shortcut is to use IF...THEN...ELSE statements to combine lines. For example, if you have:

```
10 IFF=0THENB=105
```

```
20 IFE=0THENB=63
```

```
30 IFD=0THENB=21
```

since these three statements are mutually exclusive, that is only one of the three will be true at any given time, you could combine them into one long statement like this:

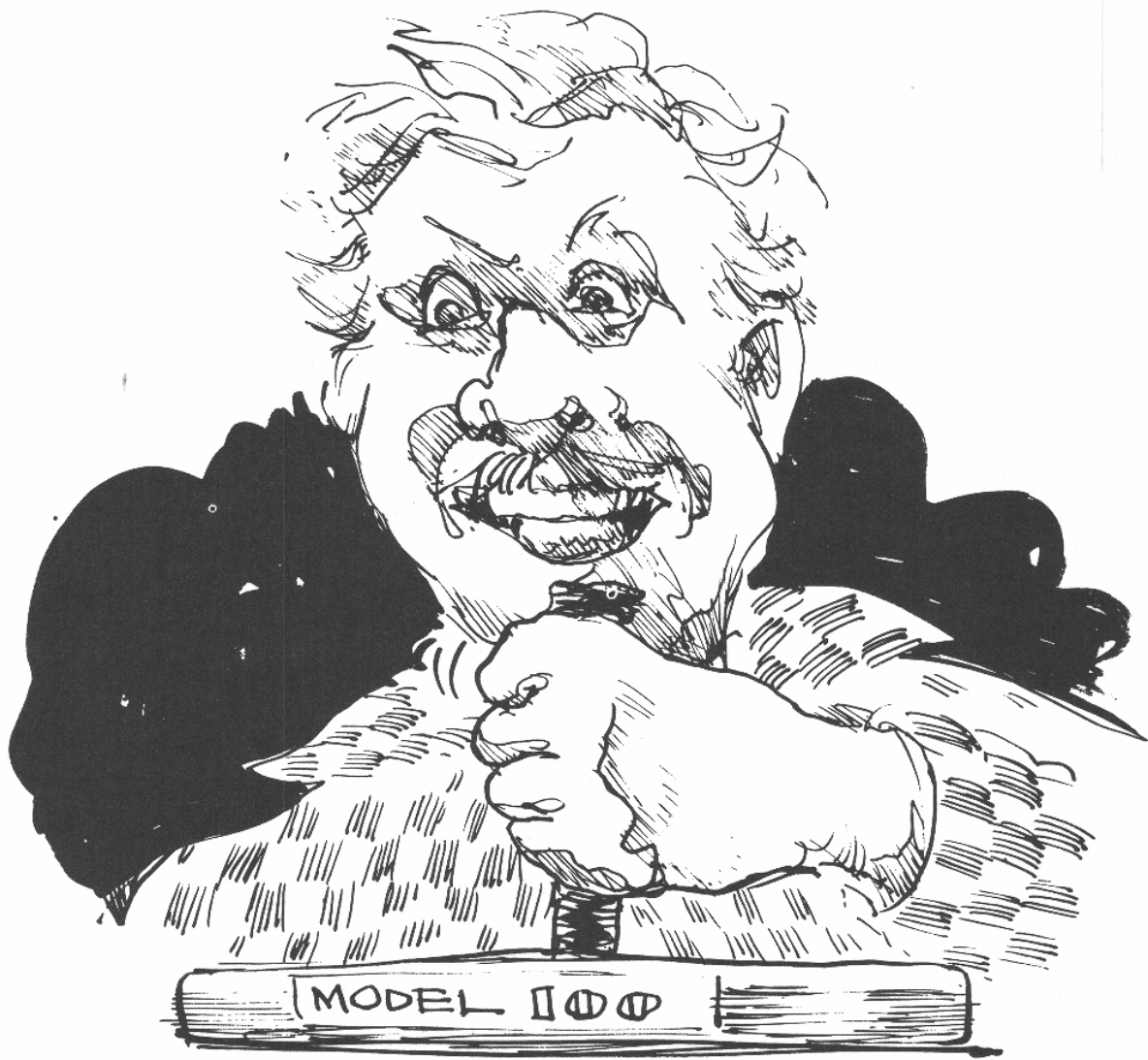
```
10 IFF=0THENB=105ELSEIFE=0THENB=63ELSEIFD=0THENB=21
```


You have to be sure that the conditions in the IF...THEN...ELSE statement are mutually exclusive or you'll end up changing a variable when it shouldn't be changed, or branching to a line when no branching is needed.

Another trick is to reduce the length of messages and prompts to the program user; i.e., printing "Press ENTER" instead of "Press ENTER to continue," a difference of thirteen bytes. Or you can remove user instructions like those in REVERS.

With these alterations, the size of a program can be considerably reduced, sometimes by hundreds of bytes.

The first program in this section is the perennial favorite, Tic-tac-toe; followed by Revers, a number-manipulation game. Then there's Mastermind, Mimic, Hunt the Wumpus and Invaders.



TICTAC

Tic-tac-toe on the Model 100.

This program uses the four arrow keys at the top right of the keyboard for inputting your moves. When you run the program, it waits for about one second for you to make a move by pressing one of the arrow keys. If you don't, the computer makes its move first, playing X. If you move first, you get the X. To select your move, use the arrow keys to maneuver the graphic dot around the squares of the Tic-Tac-Toe board, when you reach the square you want, press the ENTER key. The location of the dot determines where your piece will be placed on the display.

After your move, the program selects a square for its move, places its piece, and then displays "Your Move," putting the graphic dot in an unoccupied square.

After each move, both yours and the computer's, the program checks the board to see if there're any squares left open. If not, it jumps to the "Who won?" routine.

If there is a square open, the program makes a move if it's the computer's move, or places the graphics dot in an open square if it's your move.

When moving the dot around the display, you'll notice that the program won't even let the dot appear in an occupied square, skipping over it instead.

The logic for the program's move is very straightforward. First it checks to see if your next move will win you the game, if so, it blocks your move by taking that square for itself. If you don't have a winning move available, it checks the board for a possible move that'll let it win the game. If that isn't possible, it makes a move at random to one of the empty squares. By the way, if you don't take the center square for your piece as your first move, the computer takes it.

If you find the game too simple, you can develop logic for the opening moves of the game by programming in possible moves at each of the corners.

Speaking of logic, the program usses a 3 x 3 array to store the board, placing a one in each square that it takes, and placing a -1 in each that you take.

Program Listing for TICTAC

```
10 REM 2784 bytes free
20 MAXFILES=0: CLEAR256: CLS
25 FOR I=1 TO VAL(RIGHT$(TIME$,2)): B=RND(1)
   : NEXT
30 PRINT TAB(25) "TIC TAC TOE": DIM A(2,2)
35 PRINT TAB(21) "Use the arrow keys"
40 PRINT TAB(21) "to move around."
50 PRINT TAB(21) "Press ENTER for"
60 PRINT TAB(21) "your move."
70 PRINT TAB(24) "If you don't"
80 PRINT TAB(21) "move first, I will.";
90 PRINT@305, "Your move.";
100 LINE(42,0)-(42,63)
110 LINE(84,0)-(84,63)
120 LINE(6,21)-(120,21)
130 LINE(6,42)-(120,42)
140 A$=INKEY$: A=A+1: IFA$="" AND A<500 THEN 140
150 IFA$<>"" AND (A$<CHR$(28) OR A$>CHR$(31)) THEN 140
160 B=63: C=32
170 PRINT@221, STRING$(19,32);
180 PRINT@261, STRING$(19,32);
190 IFA=500 THEN A(1,1)=1: A=1: GOTO 310 ELSE A=0
200 PSET(B,C): PSET(B-1,C-1): PSET(B-1,C):
   PSET(B,C-1): PRINT@305, "Your move.";

210 A$=INKEY$: IFA$="" THEN 210
220 PRINT@305, STRING$(10,32);
230 PRESET(B,C): PRESET(B-1,C-1): PRESET(B-1,C):
   PRESET(B,C-1): A=0
240 IFA$=CHR$(28) THEN B=B+42: IF B>105 THEN B=21:
   C=C+21: IF C>53 THEN C=11
250 IFA$=CHR$(29) THEN B=B-42: IF B<21 THEN B=105:
   C=C-21: IF C<11 THEN C=53
260 IFA$=CHR$(30) THEN C=C-21: IF C<11 THEN C=53:
   B=B-42: IF B<21 THEN B=105
270 IFA$=CHR$(31) THEN C=C+21: IF C>53 THEN C=11:
   B=B+42: IF B>105 THEN B=21
280 IFA(INT(B/42),INT(C/21))<>0 THEN 240
290 IFA$<>CHR$(13) THEN 200
```

```

300 A(INT(B/42),INT(C/21))=-1
310 IFXTHENGOSUB980ELSEGOSUB940
320 GOSUB810:REM Test for end of game
330 IFA=0THENA$=CHR$(28):GOTO240
340 REM Computer's move
350 GOSUB370
360 A(INT(B/42),INT(C/21))=1:GOTO310
370 IFA(1,1)=0THENB=63:C=32:RETURN
380 B=0:D=1:E=1:F=0
390 IFA(B,0)=DANDA(B,1)=EANDA(B,2)=FTHEN
    440
400 IFB<2THENB=B+1:GOTO390
410 B=0:IFF=0ANDE=1THENE=0:F=1:GOTO390
420 IFE=0ANDD=1THEND=0:E=1:GOTO390
422 IFD=0ANDF=1THEND=-1:E=-1:F=0:GOTO390
424 IFF=0THENE=0:F=-1:GOTO390
426 IFE=0THEND=0:E=-1:GOTO390
430 GOTO490
440 IFF=0THENC=53
450 IFE=0THENC=32
460 IFD=0THENC=11
470 B=B*42+21
480 RETURN
490 C=0:D=1:E=1:F=0
500 IFA(0,C)=DANDA(1,C)=EANDA(2,C)=FTHEN
    550
510 IFC<2THENC=C+1:GOTO500
520 C=0:IFF=0ANDE=1THENE=0:F=1:GOTO500
530 IFE=0ANDD=1THEND=0:E=1:GOTO500
532 IFD=0ANDF=1THEND=-1:E=-1:F=0:GOTO500
534 IFF=0THENE=0:F=-1:GOTO500
536 IFE=0THEND=0:E=-1:GOTO500
540 GOTO600
550 IFF=0THENB=105
560 IFE=0THENB=63
570 IFD=0THENB=21
580 C=C*21+11
590 RETURN
600 D=1:E=1:F=0
610 IFA(0,0)=DANDA(1,1)=EANDA(2,2)=FTHEN
    650
620 IFF=0ANDE=1THENE=0:F=1:GOTO610
630 IFE=0ANDD=1THEND=0:E=1:GOTO610
632 IFD=0ANDF=1THEND=-1:E=-1:F=0:GOTO610
634 IFF=0THENE=0:F=-1:GOTO610
636 IFE=0THEND=0:E=-1:GOTO610
640 GOTO690
650 IFF=0THENB=105:C=53

```

```

660 IFE=0THENB=63:C=32
670 IFD=0THENB=21:C=11
680 RETURN
690 D=1:E=1:F=0
700 IFA(0,2)=DANDA(1,1)=EANDA(2,0)=FTHEN
    740
710 IFF=0ANDE=1THENE=0:F=1:GOTO700
720 IFE=0ANDD=1THEND=0:E=1:GOTO700
722 IFD=0ANDF=1THEND=-1:E=-1:F=0:GOTO700
724 IFF=0THENE=0:F=-1:GOTO700
726 IFE=0THEND=0:E=-1:GOTO700
730 GOTO780
740 IFF=0THENB=105:C=11
750 IFE=0THENB=63:C=32
760 IFD=0THENB=21:C=53
770 RETURN
780 B=INT(RND(1)*3):C=INT(RND(1)*3)
790 IFA(B,C)<>0THEN780
800 B=B*42+21:C=C*21+11:RETURN
810 FORI=0TO2:T=A(I,0)+A(I,1)+A(I,2):IFA
    BS(T)=3THEN880ELSENEXT
820 FORI=0TO2:T=A(0,I)+A(1,I)+A(2,I):IFA
    BS(T)=3THEN880ELSENEXT
830 T=A(0,0)+A(1,1)+A(2,2):IFABS(T)=3THE
    N880
840 T=A(2,0)+A(1,1)+A(0,2):IFABS(T)=3THE
    N880
850 Z=0:FORI=0TO2:FORJ=0TO2:IFA(I,J)=0TH
    ENZ=1
860 NEXT:NEXT:IFZTHENRETURN
870 PRINT@261,"Nobody won!";
880 IFT=3THENPRINT@261,"The Computer Win
    s!"
890 IFT=-3THENPRINT@263,"You win!"
900 PRINT@305,"Play again?";
910 A$=INKEY$:IFA$="Y"ORA$="y"THEN20
920 IFA$<>"N"ANDA$<>"n"THEN910
930 END
940 X=1:IFA=1THENA=0ELSEA=1
950 LINE(B-6,C-6)-(B+6,C+6)
960 LINE(B-6,C+6)-(B+6,C-6)
970 RETURN
980 X=0:IFA=0THENA=1ELSEA=0
990 LINE(B-6,C-6)-(B+6,C+6),1,B
1000 RETURN

```

REVERS

A simple analytical game.

Revers is a simple analytical game. You start off with a series of numbers, up to a maximum of twenty-six. These numbers are scrambled in a random order, your goal is to restore them into counting order, with the lowest number first.

You do this by selecting groups of numbers and reversing their order, starting with the first number on the display. Confused? Well, for example, let's say you have eight numbers. The display would be:

```
1 2 3 4 5 6 7 8
8 7 6 5 4 1 3 2
```

(The top row of numbers are reference numbers, the bottom row are the numbers you're trying to sort.) In this case your first reversal would be 8:

```
1 2 3 4 5 6 7 8
2 3 1 4 5 6 7 8
```

Your second reversal would be 2:

```
1 2 3 4 5 6 7 8
3 2 1 4 5 6 7 8
```

Your third reversal would be 3:

```
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
```

And you'd win in three moves! Simple? Of course it is, until you try it yourself.

Program Listing for REVERS

```
10 REM 1597 bytes
20 CLS: CLEAR 256: MAXFILES=0
30 PRINT
40 PRINT
50 PRINT "THE GAME OF REVERS ": PRINT: PRINT
   T: PRINT
60 INPUT "Do you want instructions (Y/N) "
   ; A$: IFA$="Y" OR A$="y" THEN GOSUB 440
70 CLS: IFA$ <> "N" AND A$ <> "n" THEN 20
80 INPUT "How many numbers do you want to
   reverse ---- "; R
90 IFR > 26 THEN PRINT "Maximum number is 26.
   ": GOTO 80
100 IFR=0 THEN END
110 DIM REV%(R, 2)
120 FOR X=0 TO R
130 REV%(X, 1)=X
140 N=N+1
150 NEXT
160 FOR X=0 TO R*10
170 R1=INT(RND(1)*R): R2=INT(RND(1)*R)
180 IFR1=0 OR R2=0 THEN 170
190 A1=REV%(R1, 1): A2=REV%(R2, 1)
200 REV%(R1, 1)=A2: REV%(R2, 1)=A1
210 NEXT
220 CLS
230 S=0: N=0: FOR X=1 TO R
240 N=N+1
250 PRINT @((N-1)*3)+S, USING "##"; X;
260 PRINT @((N-1)*3)+S+40, USING "##"; REV%(
   X, 1)
270 IF X=13 THEN S=S+120: N=0
280 NEXT
290 INPUT "How many do I reverse"; NUMBER:
   IF NUMBER > R THEN PRINT "Too high. ": GOTO
   290
300 M=M+1
310 IF NUMBER=0 THEN END
320 N=0: FOR X=NUMBER TO 1 STEP -1
330 N=N+1
340 REV%(X, 2)=REV%(N, 1)
350 NEXT
360 FOR X=1 TO NUMBER
370 REV%(X, 1)=REV%(X, 2): NEXT
```

```

380 FORX=1TOR:IFREV%(X,1)=XTHEN:NEXT:GOT
    O400
390 GOTO220
400 CLS:PRINT:PRINT:PRINT"You did it in"
    ;M;"moves!"
410 PRINT:INPUT"Do you want to try again
    (Y/N)";A$
420 IFLEFT$(A$,1)="Y"ORLEFT$(A$,1)="y"TH
    EN20
430 IFLEFT$(A$,1)="N"ORLEFT$(A$,1)="n"TH
    ENENDELSE410
440 CLS:PRINT"    The game of Revers  is
    a game of"
450 PRINT"skill. You start by telling me
    how many"
460 PRINT"numbers you want, then you mus
    t put"
470 PRINT"them in numerical order (1,2,3
    ,...etc.)"
480 PRINT"    You do this by reversing th
    e numbers"
490 PRINT"at the front of the string."
500 GOSUB580:CLS
510 PRINT"    For example, if you have: 65
    4321789"
520 PRINT"and you reverse 6, you get: 12
    3456789"
530 PRINT"and you win."
540 PRINT"    You should be able to do it
    in 2*N-3"
550 PRINT"moves (N=number of numbers to
    reverse)."

```

Ω

MASTER

Play Mastermind on the Model 100.

Mastermind is another popular thinking game. The computer selects a five-digit number and you try to guess what that number is. This isn't as impossible as it sounds, since the program gives you clues each time you guess at the number. However, the clues are coded.

The codes are very simple: the program uses a two-digit number to tell you how many of the digits in your number guess match the digits in its number. The ones position of the number is used to tell you how many you matched. If you accidentally get one or more of the digits in your number in the exact same position as the digit in the program's number, then the tens digit is used to tell you how many you exactly matched.

For example, if the clue is 03, then three of the numbers in your guess are correct. If the clue is 30, then three of the numbers in your guess are the same as the program's number and all three are correctly positioned. The clue 12 means that three of the digits of your guess are correct, and that one of them is in the correct position.

As a final note, the program's number can have the same digit more than once, i.e., the number could be 77877. And if you want to cheat, you can see the program's number by simply pressing the ENTER key instead of typing in a number guess.

Program Listing for MASTER

```
10 REM 1567 bytes
20 MAXFILES=0: CLEAR 256
30 GOSUB 670
40 DIM A(5), G(5), T A(5), S(5), T G(5)
50 PRINT
```

```

60 PRINT"Do you want instructions (Y/N)"
70 A$=INKEY$:R=RND(1):IFA$=""THEN70
80 IFA$="Y"ORA$="Y"THEN460ELSEIFA$<>"N"A
    NDA$<>"n"THEN20
90 GOSUB670
100 FORX=1TO5
110 A(X)=INT(RND(1)*9)
120 NEXT
130 PRINT
140 C=0
150 PRINT"Your five digit guess?"
160 N=0
170 PRINT"Guess number"C
180 N=N+1
190 A$=INKEY$
200 IFA$=""ORA$>CHR$(65)THEN190
210 IFA$=CHR$(13)THENPRINT:FORY=1TO5:PRI
    NTA(Y);:NEXT:PRINT:GOTO190
220 G(N)=VAL(A$)
230 PRINTG(N);
240 IFN<5THEN180
250 T=0
260 FORX=1TO5
270 TA(X)=0
280 TG(X)=0
290 IFA(X)=G(X)THENTG(X)=1:TA(X)=1:T=T+1
    0
300 NEXT
310 FORB1=1TO5
320 FORB2=1TO5
330 IFTA(B1)=1ORTG(B2)=1THEN350
340 IFA(B1)=G(B2)THENT=T+1:TA(B1)=1:TG(B
    2)=1
350 NEXT
360 NEXT
370 PRINTTAB(35)USING"Clue : ##";T
380 C=C+1
390 IFT<>50THEN160
400 PRINT
410 PRINT"You WIN!"
420 PRINT"Play again (Y/N)"
430 A$=INKEY$:IFA$=""THEN430
440 IFA$<>"Y"ORA$="Y"THEN90
450 END
460 GOSUB670
470 PRINT"    In this game, I, the comput
    er, will"
480 PRINT"pick a five digit number.  You

```

```

      must try"
490 PRINT"to guess the number by the clues that"
500 PRINT"I'll give you.
510 PRINT" The clues are in the form of
      a number.";
520 PRINT"When the number is greater than or equal";
530 GOSUB650
540 PRINT"to ten, then you have at least
      one num-"
550 PRINT"ber in the right spot, i.e.; if the clue";
560 PRINT"is 23, then you have two of the numbers"
570 PRINT"in the right spot and you have correct-"
580 PRINT"ly guessed the other three numbers, but"
590 PRINT"they're in the wrong positions

600 GOSUB650
610 PRINT" Numbers can be used more than once."
620 PRINTTAB(15);"GOOD LUCK"
630 PRINT"If you press ENTER, I'll display my num-"
640 PRINT"ber for you, but that's cheating.":GOSUB650:GOTO90
650 PRINT@287,"Press ENTER to continue";
660 IFINKEY$<>CHR$(13)THEN660
670 CLS:PRINTTAB(13)"Mastermind ":RETURN

```


MIMIC

Test your memory.

How good is your memory? Can you accurately remember a series of numbers the first time you hear them? Are you interested in improving your memory? If your answer to any of these is yes, you'll like MIMIC, a memory game.

Mimic's layout is similar to a tic-tac-toe board, three rows of three boxes. The boxes are numbered left to right, bottom to top, just like the numeric pad on the Model 100 keyboard. That is, the bottom row is numbered 1, 2 and 3, the middle row is 4, 5 and 6, and the top row is 7, 8 and 9.

To start the game, press ENTER in response to the prompt. The program will beep and a graphics block will blink beside the chosen character. Press the numeric key that matches the character chosen.

The program will now select another character, blink and beep beside it, then beside the first character. You must press the numbers of both characters, in the order chosen by the program. Each time you successfully complete the sequence, another number is added to it, until you fail.

The better your memory, the longer you can go before failing. When playing the game, don't forget to press the "NUM" key to use the numeric pad.

Program Listing for MIMIC

```
10 REM 855 bytes
20 MAXFILES=0: CLEAR40:CLS
30 PRINTTAB(15)"Mimic"
40 DIMV(3,3),P(3,3),N(20,1)
50 PRINT@40,"Press any key to start."
60 A$=INKEY$:R=RND(1):IFA$=""THEN60
70 PRINT@40,STRING$(30,32):N=9
```

```

80 FORY=1TO3:FORX=3TO1STEP-1
90 V(X,Y)=N:N=N-1
100 P(X,Y)=(Y*80)+(X*3)
110 NEXT:NEXT
120 FORY=1TO3:FORX=1TO3
130 PRINT@P(X,Y),V(X,Y);
140 NEXT:NEXT
150 VN=0:DF=30
160 E=0
170 VN=VN+1
180 N(VN,0)=INT(RND(1)*3)+1
190 N(VN,1)=INT(RND(1)*3)+1
200 FORX=1TOVN
210 NX=N(X,0):NY=N(X,1)
220 PRINT@P(NX,NY),CHR$(239);
230 S=V(NX,NY)*1700
240 SOUND S,DF
250 PRIOT@P(NX,NY)," ";
260 NEXT
270 DF=DF-2
280 IFDF<5THENDF=5
290 FORX=1TOVN
300 NX=N(X,0):NY=N(X,1)
310 LN=V(NX,NY)
320 FORL=1TODF*1000
330 A$=INKEY$
340 IFA$<>" "THENL=DF*1000
350 NEXT
360 IFA$=" "THENA$="0"
370 V=VAL(A$)
380 S=LN*1700
390 SC=VN*100
400 IFV<>LNTHENX=VN:E=1:GOTO440
410 PRINT@P(NX,NY),CHR$(239);
420 SOUNDS,3
430 PRINT@P(NX,NY)," ";
440 NEXT
450 IFE<>1THEN170
460 PRINT@280,"Your score was"SC;
470 SOUND100,20
480 FORL=1TO1000:NEXT:PRINT@280,STRING$(
    30,32);
490 GOTO50

```

Ω

WUMPUS

A simple adventure game.

This is a simple adventure program. You are a hunter seeking to destroy all the evil Wumpii in their giant lair. You have only a few arrows to do the job, and no light to see by. You must feel your way around the complex cave system by listening carefully, sniffing (the Wumpus has a very distinctive odor), and staying alert for the drafts that indicate you're close to one of the many bottomless pits.

To kill a Wumpus, you must shoot straight into his cave. If you shoot into a nearby cave by mistake, you'll awaken the Wumpus and he'll wander off into another cave. Walking into a Wumpus cave will wake him up, and you'll become one of his many midday snacks.

Also, be on the alert for the giant bats. If you accidentally wander into one of their caves, they'll pick you up and carry you off into the cave system, dropping you at random and leaving you lost.

Good luck, intrepid explorer!

When starting the game, you're asked for the number of caves. The number of caves will determine how many arrows you have, and the numbers of pits, bats and Wumpii. The number of caves must be divisible by five.

Oh, as a final note, you can increase your arrow supply by making the variable AR larger (line 430). And you can get a map of the cave system by typing a "C" in response to the "Move or Shoot prompt," but that would be cheating. Make your own map as you wander through the cave.

Program Listing for WUMPUS

```
10 REM 3526 BYTES  
20 GOTO80
```



```

30 GO(1)=C(XL,YL-1)
40 GO(2)=C(XL+1,YL)
50 GO(3)=C(XL,YL+1)
60 GO(4)=C(XL-1,YL)
70 RETURN
80 MAXFILES=0:CLS:CLEAR100
90 PRINTTAB(13)"Super Wumpus":PRINT
100 YL=VAL(RIGHT$(TIMES,2)):FORX=1TOYL:X
    L=RND(1):NEXT
110 PRINT"    The number of caves must be
    divisible by five."
120 PRINT
130 INPUT"Number of caves";NC
140 IFNC/5<>INT(NC/5)THEN130
150 NR=NC/5
160 DIMC(6,NR+1),T(6,NR+1),GO(4),TT(4)
170 CLS:PRINT"Setting caves now."
180 C=1
190 PRINT"Step 1"
200 FORY=1TONR
210 FORX=1TO5
220 C(X,Y)=C
230 C=C+1:NEXT:NEXT
240 PRINT"Step 2"
250 FORL=1TONC*3
260 XR=INT(RND(1)*5)+1:YR=INT(RND(1)*NR)
    +1
270 IFRX=1ANDYR=1THEN260
280 R1=INT(RND(1)*5)+1:R2=INT(RND(1)*NR)
    +1
290 IFR1=1ANDR2=1THEN280
300 A=C(XR,YR):B=C(R1,R2)
310 C(XR,YR)=B:C(R1,R2)=A
320 NEXT:PRINT"Step 3"
330 FORL=1TONC*.08
340 GOTO380
350 RX=INT(RND(1)*5)+1:RY=INT(RND(1)*NR)
    +1:IFRX=1ANDRY=1THEN350
360 IFT(RX,RY)<>0THEN350
370 RETURN
380 GOSUB350:T(RX,RY)=1:REM    SET WUMPUS
390 IFL>3THEN420
400 GOSUB350:T(RX,RY)=2:REM    SET BATS
410 GOSUB350:T(RX,RY)=3:REM    SET PITS
420 NEXT
430 AR=INT(NC*.1)
440 GOTO580
450 REM          PRINT BOARD

```

```

460 CLS:FORYP=1TONR
470 FORXP=1TO5
480 PRINTUSING" ##";C(XP,YP);
490 NEXT:PRINT" ";:FORXT=1TO5
500 IFT(XT,YP)=1THENPRINT" W";:GOTO540
510 IFT(XT,YP)=2THENPRINT" B";:GOTO540
520 IFT(XT,YP)=3THENPRINT" P";:GOTO540
530 PRINTUSING" ##";T(XT,YP);
540 NEXT
550 PRINT:NEXT
560 FORL=1TO1000:NEXT
570 RETURN
580 REM----- START, GAME HERE -----
590 X=1:Y=1
600 CLS:PRINT"You have";AR;"arrows left"
610 PRINT"You are in cave #";C(X,Y)
620 TR=T(X,Y):IFTR=0THEN640
630 ONTRGOTO1370,1400,1450
640 PRINT"There are tunnels to caves:";
650 XL=X:YL=Y:GOSUB30
660 FORL=1TO4:IFGO(L)=0THEN680
670 PRINTUSING" ###";GO(L);
680 NEXT
690 PRINT
700 TT(1)=T(X,Y-1)
710 TT(2)=T(X+1,Y)
720 TT(3)=T(X,Y+1)
730 TT(4)=T(X-1,Y)
740 TW=0:TB=0:TP=0
750 FORL=1TO4
760 IFTT(L)=1THENTW=1
770 IFTT(L)=2THENTB=1
780 IFTT(L)=3THENTP=1
790 NEXT
800 IFTW=1THENPRINT"I smell a Wumpus!"
810 IFTB=1THENPRINT"I hear bat wings!"
820 IFTP=1THENPRINT"I feel a draft."
830 PRINT"Do you want to <M>ove or <S>ho
    ot arrow?"
840 A$=INKEY$:IFA$=""THEN840
850 IFA$="M"ORA$="m"THEN890
860 IFA$="S"ORA$="s"THEN1080
870 IFA$="C"THENGOSUB450:GOTO600
880 GOTO840
890 REM--- MOVE ----
900 INPUT"New Cave #";MC
910 D=0
920 FORL=1TO4

```

```

930 IFGO(L)=MCTHEND=L
940 NEXT
950 IFD=0THENPRINT"You can't go there. Y
    ou can go":FORTL=1TO4:PRINTGO(TL);:
    NEXT:GOTO900
960 IFD<>1THEN990
970 IFY=1THENY=NRELSEY=Y-1
980 GOTO600
990 IFD<>2THEN1020
1000 IFX=5THENX=1ELSEX=X+1
1010 GOTO600
1020 IFD<>3THEN1050
1030 IFY=NRTHENY=1ELSEY=Y+1
1040 GOTO600
1050 IFD<>4THEN600
1060 IFX=1THENX=5ELSEX=X-1
1070 GOTO600
1080 REM--- SHOOT -----
1090 PRINT"Shoot into which cave";
1100 XL=X:YL=Y:GOSUB30
1110 INPUTSC
1120 IFSC=0THEN600
1130 XL=X:YL=Y:GOSUB30
1140 C=0:FORL=1TO4:IFGO(L)=SCTHENC=1
1150 NEXT:IFC=0THENPRINT"You can't shoot
    there.":GOTO1090
1160 AR=AR-1
1170 FORXL=X-1TOX+1:FORYL=Y-1TOY+1
1180 IFC(XL,YL)=SCTHENXF=XL:YF=YL
1190 NEXT:NEXT
1200 IF T(XF,YF)=1THEN1220
1210 GOTO1480
1220 T(XF,YF)=0:PRINT"You got him!"
1230 WC=0:FORWX=1TO5:FORWY=1TONR
1240 IFT(WX,WY)=1THENWC=WC+1
1250 NEXT:NEXT
1260 IFWC=0THEN1610
1270 PRINT"There ";
1280 IFWC=1THENPRINT"is";ELSEPRINT"are";
1290 PRINTWC;"Wumpus left."
1300 FORL=1TO1000:NEXT
1310 IFAR>00THEN600
1320 PRINT"You're out of arrows, you'll
    die in"
1330 PRINT"the caves."
1340 PRINT"Play another game?"
1350 A$=INKEY$:IFA$=""THEN1350
1360 IFA$="y"ORA$="Y"THENRUNELSEEND

```

```

1370 REM >>>>> WUMPUS CAVE <<<<<<<
1380 PRINT"Wumpus cave. He wakes up and
      eats you."
1390 GOTO1340
1400 REM//////// BAT CAVE //////////
1410 CLS:PRINT:PRINT:PRINT"A super bat!"
1420 PRINT"AAAAAAAAAAHHHHHHHHHHH !!!!!"
1430 FORL=1TO200:NEXT
1440 X=INT(RND(1)*5):Y=INT(RND(1)*:NR):G
      OTO600
1450 REM+++++ PIT CAVE ++++++
1460 PRINT"You fall into a bottomless pi
      t and die."
1470 GOTO1340
1480 REM:::::::::: WUMPUS MOVES::::::::::
1490 CLS:PRINT"You miss the Wumpus. He
      wakes up, and"
1500 PRINT"moves to another cave.":IFAR=
      0THEN1320
1510 FORXL=X-1TOX+1:FORYL=Y-1TOY+1
1520 IFT(XL,YL)=1THENWX=XL:WY=YL
1530 NEXT:NEXT:FORXL=1TO1000:NEXT
1540 IFWX=0ORWY=0THEN600
1550 XL=WX:YL=WY:GOSUB30
1560 R=INT(RND(1)*4):WM=GO(R)
1570 IFWM=0ORWM=C(X,Y)THEN1560
1580 XF=0:YF=0:FORXL=WX-1TOWX+1:FORYL=WY
      -1TOWY+1
1590 IFC(XL,YL)=WMTHENT(WX,WY)=0:T(XL,YL
      )=1
1600 NEXT:NEXT:GOTO600
1610 CLS:PRINT"YOU WIN! You got them al
      l!"
1620 GOTO1340

```


INVDRS

A Model 100 takeoff of Invaders games.

This is a takeoff of the popular Invaders game. Since you can't peek the display to find graphic points on the Model 100, this program uses normal characters.

You are the right caret (>) in the upper-left corner of the display. The invaders are attacking from the upper-right corner, coming straight at you.

The invaders are represented by the digits 0-9. To destroy an invader, you must type his number. If you hit the wrong key, you get points subtracted from your score, the correct number (with your response) is displayed in place of your ">" symbol, and the computer beeps at you.

As you play the game, the invaders begin to speed up. Eventually, they'll be moving so fast only a touch typist could keep them away.

As an aside, this game makes a fun way to learn the positions of the numbers on the Model 100 numeric pad. I tried it and it works quite well.

Program Listing for INVDRS

```
10 REM 1103 bytes
20 CLS:CLR100
30 PRINTTAB(10)"INVADERS":PRINT
40 PRINT"Do you need instructions (Y/N) "
50 A$=INKEY$:R=RND(1):IFA$=""THEN50
60 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
70 IFA$="Y"THEN440
80 IFA$<>"N"THEN50
90 CLS
100 DF=200
110 PO=39
120 A$=""
130 R=INT(RND(1)*10)
140 A$=A$+STR$(R)
```

```

150 PO=PO-2
160 PRINT@0,">";
170 PRINT@PO,A$;
180 IFA$=""THEN130
190 L$=MID$(A$,2,1)
200 C=0
210 FORX=1TODF
220 I$=INKEY$
230 IFI$<>L$ANDI$<>" "THENSC=SC-45:SOUND1
    500,5
240 IFI$<>" "THENCLS:PRINTI$;L$;
250 IFI$<>L$THEN310
260 X=DF
270 PO=PO+2:SC=SC+10
280 A$=RIGHT$(A$,LEN(A$)-2)
290 IFDF>50THENDF=DF-10ELSEDF=DF-1
300 C=1
310 NEXT
320 IFPO<4THEN360
330 IFDF<00THENDF=00
340 IFC=1THEN160
350 GOTO130
360 PRINT:PRINT:PRINT"*****CRASH?*****"
370 PRINT"Your score was:";SC
380 PRINT:PRINT:PRINT"Play again (Y/N)"
390 A$=INKEY$:IFA$=""THEN390
400 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
410 IFA$="Y"THEN40
420 IFA$<>"N"THEN390
430 END
440 CLS
450 PRINT"You are the > in the upper-lef
    t corner."
460 PRINT"The Invaders are the numbers c
    oming"
470 PRINT"at you from the upper-right co
    rner."
480 PRINT"To destroy them you must press
    the"
490 PRINT"number of the one leading the
    attack."
500 PRINT"Be careful, their attack speed
    s up as"
510 PRINT"you play the game."
520 PRINT@287,"Press any key to continue
    .";
530 A$=INKEY$:IFA$=""THEN 530
540 GOTO90

```

PART V

Graphics

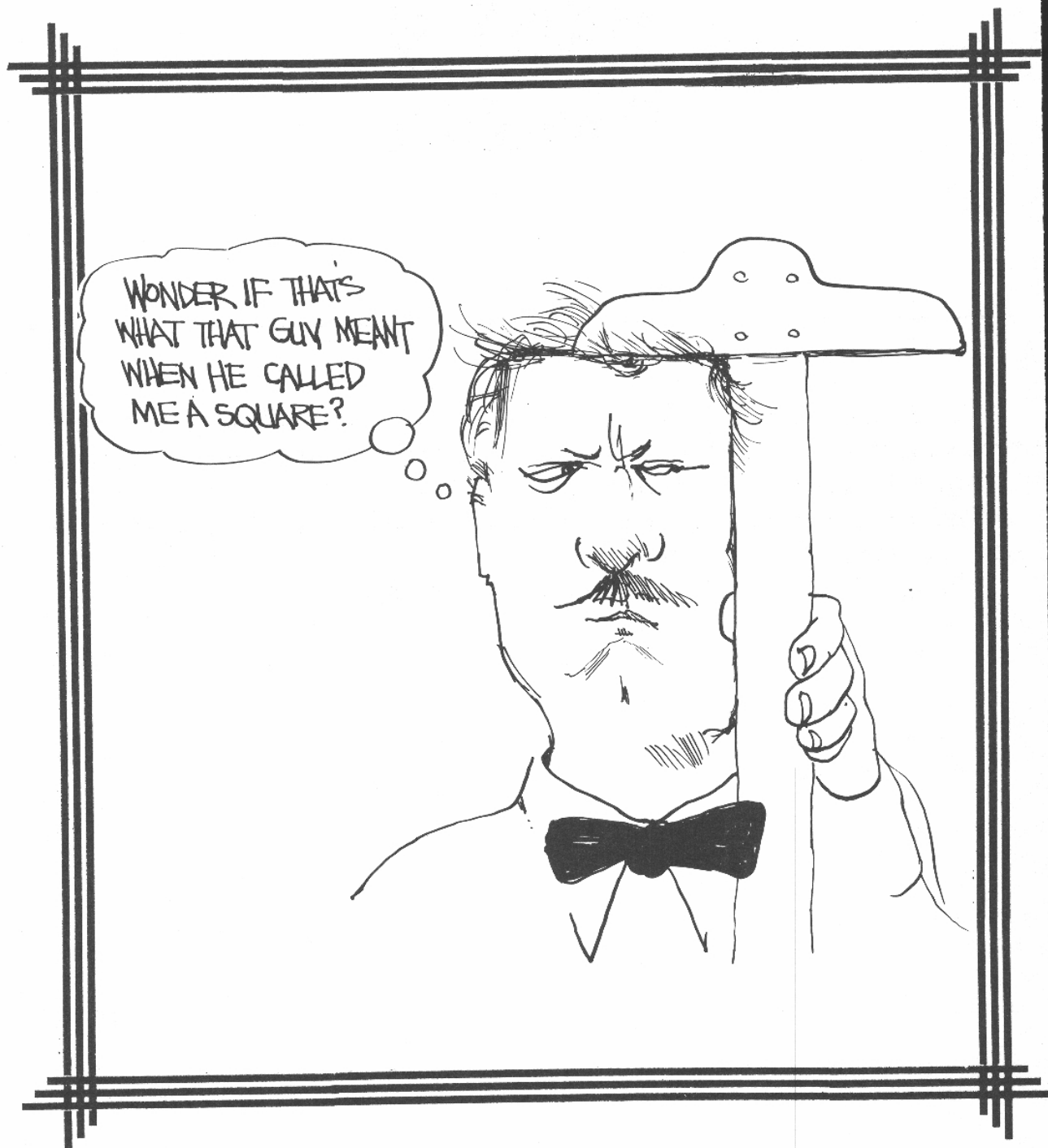
The Model 100 is capable of some very respectable graphics. Two of the programs in this section, DIPSY and ART, draw various graphic displays.

The third program, DOODLE, uses the display as a drawing board.

The next program uses your printer to prepare large continuous sheet banners for home, office or party.

Finally, there is BIORHY, a biorhythm program.

Ω



DIPSY & ART

Use math functions to draw shapes and geometrical images.

These two programs use the various math functions of the Model 100 to draw geometrical shapes on the display. These kinds of displays can easily be used to jazz up programs by just using them together with text messages.

Because of the design of the Model 100, putting text on the display after making a graphic drawing results in the text obscuring the graphics, but only for those characters in the message. The rest of the display is left unchanged.

Printing graphics on top of text messages does work, although it tends to make the text message hard to read.

Dipsy draws a series of lines on the display, mirror imaging the display from top to bottom and left to right. Because of a bug in the Model 100 random number generator, the pattern is always the same. ART asks you for a step size, then uses the trigonometric functions of the Model 100 to draw geometrical images.

Program Listing for DIPSY

```
5 REM 238 bytes
10 CLS:MAXFILES=0
20 S=0
30 X=120:Y=32
40 XR=INT(RND(120)*120)
50 YR=INT(RND(100)*32)
60 LINE(X,Y)-(XR,YR)
70 LINE(239-X,Y)-(239-XR,YR)
80 LINE(X,63-Y)-(XR,63-YR)
90 LINE(239-X,63-Y)-(239-XR,63-YR)
100 S=S+1:IFS>50THEN10
```

```
110 X=XR:Y=YR
120 GOTO40
```

Program Listing for ART

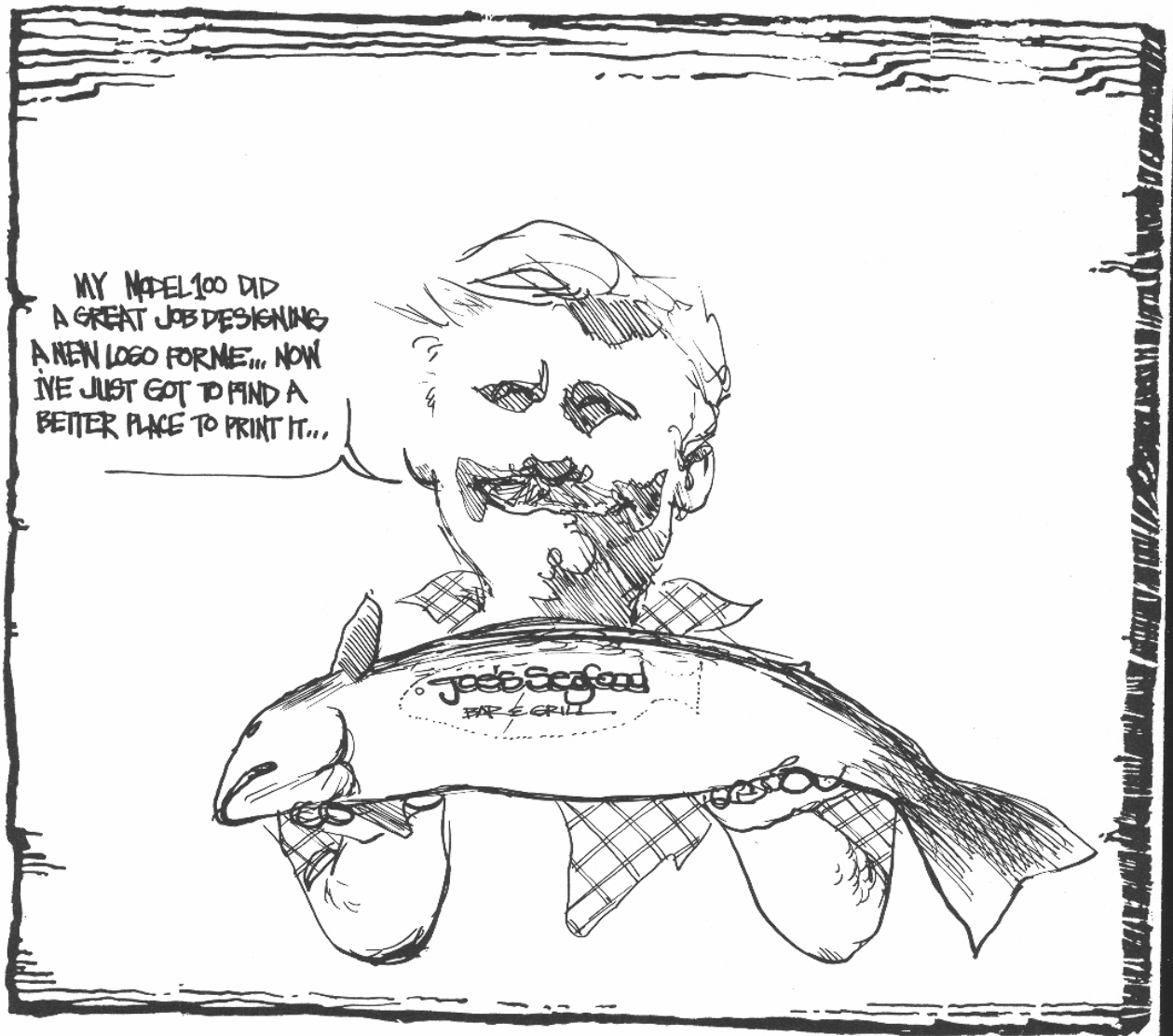
```
10 CLS:REM 909 bytes
20 INPUT"step";N
30 CLS
40 X1=119:Y1=31
50 X2=45:Y2=31
60 IF Y1<=0THEN130
70 LINE(X1,Y1)-(X2,Y2)
80 LINE(X1,63-Y1)-(X2,63-Y2)
90 LINE(239-X1,Y1)-(239-X2,Y2)
100 LINE(239-X1,63-Y1)-(239-X2,63-Y2)
110 X2=X2+N:Y1=Y1-N
120 GOTO60
130 FORL=1TO1000:NEXT
140 CLS
150 FORX=0TO239STEPN
160 LINE(0,0)-(X,63)
170 LINE(239,0)-(X,63)
180 NEXT
190 FORL=1TO1000:NEXT
200 CLS
210 FORX=0TO239STEPN
220 P=X
230 IFX>63THENP=63
240 LINE(X,63)-(239,63-P)
250 LINE(239-X,0)-(0,P)
260 NEXT
270 FORL=1TO1000:NEXT
280 CLS
290 FORX=0TO239STEPN
300 LINE(X,0)-(239-X,63)
310 NEXT
320 FORY=63TO0STEP-N
330 LINE(0,Y)-(239,63-Y)
340 NEXT
350 FORL=1TO1000:NEXT
360 CLS
370 X=0
380 XS=239/360*N
390 D=3.14159/180
400 FORA=0TO360STEPN
410 S=SIN(A*D)
```

```

420 S=(S*31)+31
430 LINE(0,0)-(X,S)
440 LINE(239,63)-(X,S)
450 X=X+XS
460 NEXT
470 FORL=1TO1000:NEXT
480 CLS
490 XC=119:YC=31:DI=15
500 FORA=0TO360STEPN*4
510 S=SIN(A*D):C=COS(A*D)
520 S=(S*DI)+YC:C=(C*DI)+XC
530 LINE(0,0)-(C,S)
540 LINE(239,0)-(C,S)
550 LINE(239,63)-(C,S)
560 LINE(0,63)-(C,S)
570 NEXT
580 FORL=1TO1000:NEXT
590 GOTO30

```

Ω



DOODLE

Use the Model 100 display as a drawing board.

Doodle turns the Model 100 display into a drawing board, letting you design and create to your heart's content.

You use the number keys, or numeric keypad, to control the direction of your pen. The SPACE BAR controls whether the pen is down and drawing or up and erasing. Press the space bar to switch modes of operation. For convenience, if you press the "R" key, then the last direction command is repeated non-stop until you press the SPACE BAR to stop it.

The only difficulty with this program is that you can't save the finished drawing for posterity. This is a limitation of the LCD display, which won't let you look at it from a program to see which dots are on or off. Unlike the other Radio Shack computers, there isn't a POINT command to give you that information.

In fact, if you turn off the computer, the graphics on the display are erased. This is because the Model 100 doesn't keep track of which dots are turned on or off. Instead it keeps track of only the text messages on the display.

Program Listing for DOODLE

```
10 REM 962 bytes
20 MAXFILES=0:CLR10:CLS:DIMV(9,2)
30 PRINTTAB(13)"Doodle Program"
40 INPUT"Do you want instructions (Y/N)"
   ;A$
50 IFA$="N"ORA$="n"THEN160
60 IFA$<>"Y"ANDA$<>"y"THEN40
70 PRINT@40,"This is a drawing program,
   use the "
```

```

80 PRINT"number keys across the top or t
   he 7 8 9"
90 PRINT"numeric keypad for the differen
   t 4 5 6"
100 PRINT"directions. The space bar lift
   s 1 2 3"
110 PRINT"the pen and erases, or drops t
   he "
120 PRINT"pen for drawing. The R key re
   peats the"
130 PRINT"last entry, space bar stops it
   . Ready?";
140 A$=INKEY$:IFA$=""THEN140
150 DATA-1,1,0,1,1,1,-1,0,0,0,1,0,-1,-1,
   0,-1,1,-1
160 CLS
170 FORX=1TO9
180 FORB=1TO2
190 READV(X,B)
200 NEXT:NEXT
210 P=1:X=120:Y=32
220 A$=INKEY$
230 PSET(X,Y)
240 PRESET(X,Y)
250 IFP=0THEN270
260 PSET(X,Y)
270 IFA$="R"ORA$="r"THENR=1
280 IFA$=" "THEN320
290 IFR=1THEN360
300 IFA$=""THEN220
310 IFA$<>" "THEN360
320 IFR=1THENR=0:GOTO220
330 IFP=1THENP=0:GOTO220
340 IFP=0THENP=1
350 GOTO220
360 V=VAL(A$)
370 IFR=1THENV=0
380 IFV<=0ORV>9THEN220
390 X=X+V(V,1):Y=Y+V(V,2)
400 IFX<0THENX=0
410 IFX>239THENX=239
420 IFY<0THENY=0
430 IFY>63THENY=63
440 O=V
450 GOTO220

```

Ω

BANNER

Create large banners on your printer.

Every computer should have a banner program available for it. A banner program is a program that writes banners on computer paper. Some programs write the banner across each page of the paper, with a maximum length determined by the width of the paper.

This program writes the banner vertically down the paper. Turning the paper sideways lets you read the message. The advantage to this method is that your message can be as long as you want it to be (well; almost, BASIC imposes a limit of 255 characters per run) assuming you have enough paper handy.

When you run the program you are asked for the size of the banner's letters. A size of one is the smallest, while letter size five almost fills the paper from edge to edge. The next prompt asks you for the paper's width in columns, so it can center the letters on the paper.

Now you can type in your message.

Currently, line 3910 makes the banner use the appropriate character of the message for each character in it. That is, the A letter is used to make the A in the banner, the \$ is used for the dollar sign character, and so forth. If you want the banner to have a uniform density, change the L\$ in line 3910 to an "X," or the character of your choice. Then the banner characters will all be formed by that character.

BANNER is a very long program. If you have an 8K computer, you won't have enough room for all the characters. By eliminating the special characters and leaving the A-Z and 0-9 characters, you can make the program short enough to fit in your 8K computer.

Hope you have fun. I've already enlivened one kid's party with a personalized message (she loved it).

Program Listing for BANNER

Note to the reader: For this program only we have doubled up the columns. Line 10 is 10 REM 4919. The 420 DATA 1000001 is a different line number. The double columns are line numbers 10 to 3700.

10 REM 4919 bytes	420 DATA1000001
20 MAXFILES=0: CLEAR256:CLS	430 DATA1001001
30 DATA A	440 DATA1111001
40 DATA1111100	450 DATA H
50 DATA0010010	460 DATA1111111
60 DATA0010001	470 DATA0001000
70 DATA0010010	480 DATA0001000
80 DATA1111100	490 DATA0001000
90 DATA B	500 DATA1111111
100 DATA1000001	510 DATA I
110 DATA1111111	520 DATA0000000
120 DATA1001001	530 DATA1000001
130 DATA1001001	540 DATA1111111
140 DATA0110110	550 DATA1000001
150 DATA C	560 DATA0000000
160 DATA0111110	570 DATA J
170 DATA1000001	580 DATA0000000
180 DATA1000001	590 DATA0100000
190 DATA1000001	600 DATA1000000
200 DATA0100010	610 DATA1000000
210 DATA D	620 DATA0111111
220 DATA1000001	630 DATA K
230 DATA1111111	640 DATA1111111
240 DATA1000001	650 DATA0001000
250 DATA1000001	660 DATA0010100
260 DATA0111110	670 DATA0100010
270 DATA E	680 DATA1000001
280 DATA1111111	690 DATA L
290 DATA1001001	700 DATA1111111
300 DATA1001001	710 DATA1000000
310 DATA1000001	720 DATA1000000
320 DATA1000001	730 DATA1000000
330 DATA F	740 DATA1000000
340 DATA1111111	750 DATA M
350 DATA0001001	760 DATA1111111
360 DATA0001001	770 DATA0000010
370 DATA0000001	780 DATA0001100
380 DATA0000001	790 DATA0000010
390 DATA G	800 DATA1111111
400 DATA0111110	810 DATA N
410 DATA1000001	820 DATA1111111

830 DATA0000010
840 DATA0000100
850 DATA0001000
860 DATA1111111
870 DATA O
880 DATA0111110
890 DATA1000001
900 DATA1000001
910 DATA1000001
920 DATA0111110
930 DATA P
940 DATA1111111
950 DATA0001001
960 DATA0001001
970 DATA0001001
980 DATA0000110
990 DATA Q
1000 DATA0111110
1010 DATA1000001
1020 DATA1010001
1030 DATA0100001
1040 DATA1011110
1050 DATA R
1060 DATA1111111
1070 DATA0001001
1080 DATA0011001
1090 DATA0101001
1100 DATA1000110
1110 DATA S
1120 DATA0100110
1130 DATA1001001
1140 DATA1001001
1150 DATA1001001
1160 DATA0110010
1170 DATA T
1180 DATA0000001
1190 DATA0000001
1200 DATA1111111
1210 DATA0000001
1220 DATA0000001
1230 DATA U
1240 DATA0111111
1250 DATA1000000
1260 DATA1000000
1270 DATA1000000
1280 DATA0111111
1290 DATA V
1300 DATA0000111

1310 DATA0011000
1320 DATA1100000
1330 DATA0011000
1340 DATA0000111
1350 DATA W
1360 DATA1111111
1370 DATA0100000
1380 DATA0010000
1390 DATA0100000
1400 DATA1111111
1410 DATA X
1420 DATA1100011
1430 DATA0010100
1440 DATA0001000
1450 DATA0010100
1460 DATA1100011
1470 DATA Y
1480 DATA0000011
1490 DATA0000100
1500 DATA1111000
1510 DATA0000100
1520 DATA0000011
1530 DATA Z
1540 DATA1100001
1550 DATA1010001
1560 DATA1001001
1570 DATA1000101
1580 DATA1000011
1590 DATA ?
1600 DATA0000000
1610 DATA0000010
1620 DATA1010001
1630 DATA0001110
1640 DATA0000000
1650 DATA !
1660 DATA0000000
1670 DATA0000000
1680 DATA1011111
1690 DATA0000000
1700 DATA0000000
1710 DATA @
1720 DATA0110010
1730 DATA1010001
1740 DATA1110001
1750 DATA1000010
1760 DATA0111100
1770 DATA #
1780 DATA0010100

1790 DATA1111111
 1800 DATA0010100
 1810 DATA1111111
 1820 DATA0010100
 1830 DATA \$
 1840 DATA0100100
 1850 DATA0101010
 1860 DATA1111111
 1870 DATA0101010
 1880 DATA0010010
 1890 DATA %
 1900 DATA1100001
 1910 DATA0110000
 1920 DATA0001100
 1930 DATA0000110
 1940 DATA1000011
 1950 DATA ^
 1960 DATA0000100
 1970 DATA0000010
 1980 DATA0000001
 1990 DATA0000010
 2000 DATA0000100
 2010 DATA &
 2020 DATA0110110
 2030 DATA1001001
 2040 DATA1010110
 2050 DATA0100000
 2060 DATA1010000
 2070 DATA >
 2080 DATA1000001
 2090 DATA0100010
 2100 DATA0010100
 2110 DATA0001000
 2120 DATA0000000
 2130 DATA '
 2140 DATA0000000
 2150 DATA0000000
 2160 DATA0000100
 2170 DATA0000011
 2180 DATA0000000
 2190 DATA]
 2200 DATA0000000
 2210 DATA1000001
 2220 DATA1111111
 2230 DATA0000000
 2240 DATA0000000
 2250 DATA)
 2260 DATA0000000

2270 DATA1000001
 2280 DATA0100010
 2290 DATA0011100
 2300 DATA0000000
 2310 DATA [
 2320 DATA0000000
 2330 DATA0000000
 2340 DATA1111111
 2350 DATA1000001
 2360 DATA0000000
 2370 DATA (
 2380 DATA0000000
 2390 DATA0011100
 2400 DATA0100010
 2410 DATA1000001
 2420 DATA0000000
 2430 DATA
 2440 DATA0000000
 2450 DATA1000000
 2460 DATA1000000
 2470 DATA1000000
 2480 DATA0000000
 2490 DATA *
 2500 DATA1001001
 2510 DATA0111110
 2520 DATA1111111
 2530 DATA0111110
 2540 DATA1001001
 2550 DATA ;
 2560 DATA0000000
 2570 DATA1000000
 2580 DATA0100100
 2590 DATA0011100
 2600 DATA0000000
 2610 DATA +
 2620 DATA0001000
 2630 DATA0001000
 2640 DATA0111110
 2650 DATA0001000
 2660 DATA0001000
 2670 DATA <
 2680 DATA0000000
 2690 DATA0001000
 2700 DATA0010100
 2710 DATA0100010
 2720 DATA1000001
 2730 DATA -
 2740 DATA0000000

2750 DATA0001000
 2760 DATA0001000
 2770 DATA0001000
 2780 DATA0000000
 2790 DATA =
 2800 DATA0000000
 2810 DATA0010100
 2820 DATA0010100
 2830 DATA0010100
 2840 DATA0000000
 2850 DATA .
 2860 DATA0000000
 2870 DATA1000000
 2880 DATA0000000
 2890 DATA0000000
 2900 DATA0000000
 2910 DATA ", "
 2920 DATA0000000
 2930 DATA1000000
 2940 DATA0110000
 2950 DATA0000000
 2960 DATA0000000
 2970 DATA /
 2980 DATA0100000
 2990 DATA0010000
 3000 DATA0001000
 3010 DATA0000100
 3020 DATA0000010
 3030 DATA " : "
 3040 DATA0000000
 3050 DATA0000000
 3060 DATA0010100
 3070 DATA0000000
 3080 DATA0000000
 3090 DATA 0
 3100 DATA0111110
 3110 DATA1010001
 3120 DATA1001001
 3130 DATA1000101
 3140 DATA0111110
 3150 DATA 1
 3160 DATA0000000
 3170 DATA1000010
 3180 DATA1111111
 3190 DATA1000000
 3200 DATA0000000
 3210 DATA 2
 3220 DATA1110010

3230 DATA1001001
 3240 DATA1001001
 3250 DATA1001001
 3260 DATA1000110
 3270 DATA 3
 3280 DATA0100010
 3290 DATA1000001
 3300 DATA1001001
 3310 DATA1001001
 3320 DATA0110110
 3330 DATA 4
 3340 DATA0001100
 3350 DATA0001010
 3360 DATA0001001
 3370 DATA1111111
 3380 DATA0001000
 3390 DATA 5
 3400 DATA0100111
 3410 DATA1000101
 3420 DATA1000101
 3430 DATA1000101
 3440 DATA0111001
 3450 DATA 6
 3460 DATA0111100
 3470 DATA1001010
 3480 DATA1001001
 3490 DATA1001001
 3500 DATA0110000
 3510 DATA 7
 3520 DATA0000001
 3530 DATA1110001
 3540 DATA0001001
 3550 DATA0000101
 3560 DATA0000011
 3570 DATA 8
 3580 DATA0110110
 3590 DATA1001001
 3600 DATA1001001
 3610 DATA1001001
 3620 DATA0110110
 3630 DATA 9
 3640 DATA1000110
 3650 DATA1001001
 3660 DATA1001001
 3670 DATA0101001
 3680 DATA0011110
 3690 DATA 00
 3700 DIMCS(63,5)

```

3710 PRINTTAB(16)"BANNER"
3720 READL$
3730 IFL$="00"THEN3800
3740 V=ASC(L$)-32
3750 C$(V,0)=L$
3760 FORX=1TO5
3770 READC$(V,X)
3780 NEXT
3790 GOTO3720
3800 INPUT"Letter size (1-5)";SL
3810 INPUT"Paper width (columns)";CO
3820 SH=SL*2
3830 TB=(CO-(SH*7))/2:GOTO3980
3840 REM---- PRINT LETTER L$
3850 V=ASC(L$)-32
3860 FORX=1TO5
3870 FORL=1TOSL
3880 FORN=1TO7
3890 CR$=MID$(C$(V,X),N,1)
3900 LPRINT TAB(TB);
3910 IFCR$="1"THENFORP=1TOSH:LPRINTL$;:N
EXT
3920 IFCR$="0"THENFORP=1TOSH:LPRINT" ";:
NEXT
3930 NEXT
3940 LPRINT
3950 NEXT
3960 NEXT
3970 RETURN
3980 REM START INPUT
3990 LINEINPUT"What message";P$
4000 FORXP=1TOLEN(P$)
4010 L$=MID$(P$,XP,1)
4020 IFASC(L$)>90THENL$=CHR$(ASC(L$)-32)
4030 IFL$<>" "THEN4050
4040 FORLF=1TOSL*4:LPRINT:NEXT:GOTO4060
4050 GOSUB3840
4060 FORLF=1TOSL
4070 LPRINT:NEXT
4080 NEXT
4090 PRINT"Another Message (Y/N)?"
4100 A$=INKEY$:IFA$=""THEN4100
4110 IFA$="N"ORA$="n"THEN4140
4120 IFA$<>"Y"ANDA$<>"y"THEN4090
4130 GOTO3800
4140 END

```

Ω

BIORHY

Chart your biorhythms with this program.

Biorhythms are a source of much curiosity for many people. This program will determine and display a graph of your emotional, physical and mental cycles for any period of twenty-one days you specify.

When you start the program, it asks for your birthday (the date of birth of the person for whom you are casting the biorhythm), month first, then day, then year. The year must be a complete figure, typing 83 for 1983 will give an error.

Next you enter the date about which you want the biorhythm drawn. The program will tell you the number of days between the two dates, then wait for you to press ENTER. When you do, it'll draw the Emotional State graph. After the graph is drawn, it waits for you to press ENTER again. Then it draws the Physical State graph, and once more waits for you to press ENTER. Finally it draws the Intellectual (Mental) State graph.

In all three graphs the big vertical line is the date you selected. The short lines are the days before and after that date. You'll notice that each graph has a different length. This is because the program draws a sinewave to fit the screen. Since each state (E, P and I) uses a different period of days this requires a different graph for each.

The critical days with biorhythms are the days where any of the three states' lines cross the horizontal center line. When the lines are at the top of the screen, your state for that line is at a high point; when the line is at the bottom of the screen, you're at a low point. If all three lines cross the center line on the same day, watch out. Similarly, if all three lines hit a low point on one day, be careful (some believers refuse to do anything on days like that, but not all of us can afford to do that).

After the graphs are drawn, you can redraw the same graph or start over and draw a new one.

Program Listing for BIORHY

```
10 REM 1953 bytes
20 CLS:MAXFILES=0:CLEAR50
30 PRINTTAB(10)"Biorhythym calculator"
40 INPUT"Your birth month";MM
50 INPUT"Day";DD
60 INPUT"Year (full year)";YY
70 IFYY<1000THENPRINT"you need to enter
    full year; i.e., 1983.":GOTO60
80 GOSUB910
90 BF=FA
100 INPUT"Present month";MM
110 INPUT"Day";DD
120 INPUT"Year (full year)";YY
130 IFYY<1000THENPRINT"Enter full year;
    i.e., 1983.":GOTO120
140 GOSUB910
150 PF=FA
160 DO=PF-BF
170 CLS:PRINT
180 PRINT"You are";DO;"days old"
190 D=3.14159/180
200 DO=DO-10
210 PRINT"Press any key to draw graphs."
220 A$=INKEY$:IFA$=""THEN220
230 P=23:E=28:I=33
240 CLS
250 PP=DO/P:EE=DO/E:II=DO/I
260 PP=PP-INT(PP):EE=EE-INT(EE)
270 II=II-INT(II)
280 II=INT((I*II)+.5)
290 PP=INT((P*PP)+.5)
300 EE=INT((E*EE)+.5)
310 EN=(EE*100)/E
320 PN=(PP*100)/P
330 IN=(II*100)/I
340 DS=(EN/100)*360
350 XO=0:YO=0
360 PRINT00,"E";
370 X=0:XI=239/E
380 LINE(0,31)-(239,31)
390 FORXX=0TO239STEPXI
400 LINE(XX,29)-(XX,33):NEXT
410 LINE(XI*10,0)-(XI*10,63)
420 FORA=DSTODS+360 STEP360/E
```



```

430 ED=(SIN(A*D)*31)+31
440 IFXO=0THEN460
450 LINE(XO,YO)-(X,ED)
460 XO=X:YO=ED
470 X=X+XI:NEXT
480 A$=INKEY$:IFA$=""THEN480
490 CLS
500 XO=0:YO=0
510 PRINT@0,"P";
520 LINE(0,31)-(239,31)
530 X=0:XI=239/P
540 DS=(PN/100)*360
550 FORXX=0TO239STEPXI
560 LINE(XX,29)-(XX,32):NEXT
570 LINE(XI*10,0)-(XI*10,63)
580 FORA=DSTODS+360STEP360/P
590 S=(SIN(A*D)*31)+31
600 IFXO=0THEN620
610 LINE(XO,YO)-(X,S)
620 XO=X:YO=S
630 X=X+XI
640 NEXT
650 A$=INKEY$:IFA$=""THEN650
660 CLS
670 DS=(IN/100)*360
680 LINE(0,31)-(239,31)
690 XI=239/I
700 FORXX=0TO239STEPXI
710 LINE(XX,29)-(XX,33):NEXT
720 LINE(XI*10,0)-(XI*10,63)
730 PRINT@0,"I";
740 X=0:XO=0:YO=0
750 FORA=DSTODS+360 STEP360/I
760 S=(SIN(A*D)*31)+31
770 IFXO=0THEN790
780 LINE(XO,YO)-(X,S)
790 XO=X:YO=S:X=X+XI
800 NEXT
810 A$=INKEY$:IFA$=""THEN810
820 CLS:PRINT"Redraw the same graph (Y/N
    )?"
830 A$=INKEY$:IFA$=""THEN830
840 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
850 IFA$="Y"THENDO=DO+10:GOTO170
860 IFA$<>"N"THEN830
870 PRINT:PRINT"Draw another graph (Y/N
    )"
880 A$=INKEY$:IFA$=""THEN880

```



```

890 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
900 IFA$="Y"THENRUNELSEIFA$="N"THENENDEL
    SE880
910 IFMM=1ORMM=2THEN940
920 FA=365*YY+DD+31*(MM-1)-INT(.4*MM+2.3
    )+INT(YY/4)-INT(3/4*(INT(YY/100)+1)
    )
930 RETURN
940 FA=365*YY+DD+31*(MM-1)+INT((YY-1)/4)
    -INT(3/4*(INT((YY-1)/100)+1))
950 RETURN

```



Ω

PART VI

Utilities

Although the design of the Model 100 is exceptional, Radio Shack did leave out one or two useful functions. This section will provide those sorely needed abilities, plus a couple more for those interested in exploring the memory of their computer.

The first two programs are called WORDCT and LENCTR. These two programs will tell you the length of your files.

For machine-language programs, CONVR will convert any number in Hexadecimal, Decimal, Octal, or Binary format to its equivalent number in the other three bases.

If you're interested in just what the memory of your computer contains, you can use MEMSCN to scan through it.

The last program is PRINT, a program for formatting your program listings.





WORDCT

Determine the word count of your files.

This program supplies a much needed function for the word processor of the Model 100; it tells you the length of your file. WORDCT scans your text file and gives a count of the actual length of your file in characters. In addition, for those writers who have to keep track of such things, it'll tell you the length of your file in words.

I wrote the program to take as little space as possible, so I could keep it in memory all the time. If you're confident of your typing (I'm not of mine), you can eliminate all the error checking at the beginning of the program (lines 20 to 40).

I use my Model 100 as a pre-processor for my text files. That is, I create the files on the Model 100, then send them to my Model I TRS-80 and use my spelling and grammar checking programs on them. Then I use my favorite word processor, NEWSRIPT, to print out the text file with justification, page numbering, etc., all those print formatting commands that aren't in the Model 100.

Because of this, I inserted line 110 in the program. This line checks each line of my text file to see if the line starts with my Model I word processor's command symbol (a period). If you want to eliminate this, change line 110 to read: "110 B=0."

Program Listing for WORDCT

```
5 REM 933
10 CLS: CLEAR 533: PRINT TAB(15) "WORDCOUNTER
   ": MAXFILES=1: FILES: INPUT "Filename";
   A$
20 IF LEN(A$) > 9 THEN PRINT "Filename too long." : GOSUB 190: RUN
```

```

30 A=INSTR(A$,"."):IFA<>0THENIFMID$(A$,A
    ,3)<>".do"ANDMID$(A$,A,3)<>".DO"THE
    NPRINT"Must be a text file (end in
    "+CHR$(34)+" ".DO"+CHR$(34)+" ".":GOSU
    B190:RUN
40 IFRIGHT$(A$,3)=" ".DO"ORRIGHT$(A$,3)=" ".
    do"THENA$=LEFT$(A$,LEN(A$)-3)
50 INPUT"Word count or Character Count (
    W/C)";C$:IFC$="c"THENC$="C"ELSEIFC$
    ="w"THENC$="W"ELSEIFC$<>"W"ANDC$<>"
    C"THEN50
60 OPEN"RAM:"+A$+".DO"FORINPUTAS1:A=0
70 IFEOF(1)THENCLOSE:PRINT@280,"Total nu
    mber of words is"A".":GOSUB190:RUN
80 LINEINPUT#1,A$:IFC$="C"THENA=A+LEN(A$
    ):IFLEN(A$)=0THENA=A+1
90 IFC$="C"THENPRINT@280,USING"Character
    Count = ##,###";A;
100 IFC$="C"THENIFEOF(1)THENPRINT@280,"f
    ile Length is"A"characters.":GOSUB1
    90:RUN:ELSE80
110 B=0:B$=MID$(A$,2,1):IFLEFT$(A$,1)="
    "AND(B$<>" "AND(B$<"0"ORB$>"9"))THE
    N70
120 IFD=1THEND=0:IFLEFT$(A$,1)<>" "THENA
    =A-1
130 IFB=255THEND=1:GOTO70
140 C=INSTR(B+1,A$," "):IFB=0ANDC=0THEN7
    0
150 IFNOTCTHEND=1
160 A=A+1:B=C:PRINT@280,USING"Word Count
    = ##,###";A;
170 IFB<255THENIFMID$(A$,B+1,1)=" "THENB
    =B+1:GOTO170
180 IFCTHEN130ELSE70
190 PRINT@287,"Press <ENTER> to continue
    ";
200 IFINKEY$<>CHR$(13)THEN200ELSERETURN

```

LENCTR

Find the number of bytes in your BASIC programs.

Another difficulty with the Model 100 is trying to determine the size of your BASIC programs. This BASIC program, LENCTR, takes care of that chore for you. (It'll even calculate its own length.)

Like WORDCT, this program uses error checking, primarily in line 50 (change it to: 50 C\$ + ".BA" to eliminate it).

Because you can't use BASIC I/O routines to read a BASIC program in memory, as you can with data files, another method has to be used. So, unlike WORDCT, LENCTR has to figure your file's length by going directly to the Model 100's directory and searching it for a match to the program name you gave it. Once it finds the program in the directory, it jumps to the address listed in the directory and starts scanning through the program to calculate its length.

This means, of course, that you don't have to save your program in ASCII (.DO) format, which would give you an incorrect file size since compressed BASIC programs are about 25% smaller than their ASCII counterparts.

This method does have a disadvantage: if the program is divided into more than one segment, the length calculated will be wrong because LENCTR won't add in the second portion of your file to its calculations. Fortunately, this only occurs when you have a large number of programs in memory simultaneously.

Program Listing for LENCTR

```
5 REM 660 bytes
10 CLS: CLEAR 256: PRINT TAB(10) "BASIC File
   length": DEF SNG A-Z: PRINT: FILES: MAXFI
   LES=1
20 PRINT: INPUT "Filename"; A$: IF LEN(A$) > 9 T
```

```

      HENPRINT"Filename too long.":GOSUB1
      20:RUN
30  FORI=1TOLEN(A$):B$=MID$(A$,I,1):IFB$=
    >"a"ANDB$<="z"THENB$=CHR$(ASC(B$)-3
    2)
40  C$=C$+B$:NEXT
50  I=INSTR(C$,"."):IFI<>0THENIFRIGHT$(C$
    ,3)<>".BA"THENPRINT"Must be a BASIC
    file.":GOSUB120:RUNELSEELSEC$=C$+"
    .BA"
60  I=63919
70  I=I+11:IFPEEK(I)=0THEN70ELSEIFI>64128
    THEN140
80  IFPEEK(I)<>128THEN70ELSEA=LEN(C$)-3:B
    =1
90  IFMID$(C$,B,1)<>CHR$(PEEK(I+2+B))THEN
    70ELSEIFB<ATHENB=B+1:GOTO90
100 GOSUB150:PRINT"File Length is"A-B+2"
    bytes.":GOSUB120:RUN
110 END
120 PRINT@288,"Press ENTER to continue";
130 IFINKEY$<>CHR$(13)THEN130ELSERETURN
140 PRINT"File not in menu.":GOSUB120:RU
    N
150 B=PEEK(I+1)+PEEK(I+2)*256:A=B
160 IFPEEK(A)=0THENRETURN
170 A=PEEK(A)+PEEK(A+1)*256:GOTO160
180 IFX$=""THENX=10ELSEX=VAL(X$):RETURNE
    LSEReturn

```

CONVR

Convert from one numbering system to another.

If you work with computers, you work with number bases other than the standard decimal, usually hexadecimal (base 16), octal (base 8), and binary (base 2). Or if you just want to learn the 8085 assembly code language, you need a program for converting from decimal (base 10) to these other bases and back.

CONVR will convert a number in any one of the four bases to its equivalent number in the other three bases. For example, to find the equivalent to 10, in decimal, in the other three, binary, octal and hexadecimal, type 10D. The program will convert this number, and display the answer for you, namely 1010 (in binary), 12 (in octal), and A (in hexadecimal).

Here's a brief lesson for the novice: binary numbers are composed of 1's and 0's, indicating only one of two possible conditions. 01 equals one, 10 equals two, 11 equals three, 100 equals four, and so forth. Octal numbers are based on the number eight. Thus eight is displayed as 10, nine is 11, ten is 12, sixteen is 20, twenty is 24. The hexadecimal number system is based on the number 16. To avoid confusion, the letters A-F are substituted for the decimal numbers 10-15. Thus 9 equals nine, A equals ten, F equals fifteen, 11 equals 17, and so forth.

For more examples on the different number bases, try running the program. When you use it, you must tell it the number base of the original number you're typing in; that is, end your number with D if the number is decimal, O if it's octal, B if it's binary, or H if it's hexadecimal.

Program Listing for CONVR

```
10 REM 1622 bytes
20 MAXFILES=0: CLEAR 256: CLS
30 PRINT TAB(4) "Binary-Octal-Hexadecimal-
    Decimal"
```



```

40 PRINTTAB(15)"Converter"
50 PRINT"End the number with a B for bin
   ary,"
60 PRINT"O for octal, H for hexadecimal,
   or
70 PRINT"D for decimal."
80 INPUT"Number";A$
90 IFA$="" THEN80
100 C$=RIGHT$(A$,1)
110 IFC$<"B" THEN80
120 IFC$>"O" THENC$=CHR$(ASC(C$)-32):GOTO
   110
130 IFC$<>"B" ANDC$<>"D" ANDC$<"H" ANDC$<>"
   O" THEN80
140 E=LEN(A$)-1:IFE=0 THEN80
150 IFC$="D" THEND=VAL(A$):GOTO400
160 FORI=1TOE+1:IFMID$(A$,I,1)>"Z" THENMI
   D$(A$,I,1)=CHR$(ASC(MID$(A$,I,1))-3
   2)
170 NEXT
180 IFC$<>"B" THEN260
190 REM Convert binary to decimal
200 C=2:M=1
210 B$=MID$(A$,M,1)
220 IFB$<>"0" ANDB$<>"1" THENPRINT"Incorre
   ct Binary input":GOTO80
230 IFM<E THENM=M+1:GOTO210
240 GOSUB630
250 GOTO400
260 IFC$<>"H" THEN350
270 REM Convert Hexadecimal to decimal
280 C=16:M=1
290 B$=MID$(A$,M,1)
300 IFB$>"F" THENPRINT"Incorrect hexadeci
   mal input":GOTO80
310 IFM<E THENM=M+1:GOTO290
320 GOSUB630
330 GOTO400
340 REM Convert Octal to decimal
350 C=8:M=1
360 B$=MID$(A$,M,1)
370 IFB$>"7" THENPRINT"Incorrect octal in
   put":GOTO80
380 IFM<E THENM=M+1:GOTO360
390 GOSUB630
400 CLS:PRINT:PRINT"Decimal"TAB(25)USING
   "###,###";D
410 IFC$="B" THENB$=LEFT$(A$,E):GOTO460

```

```

420 B=D:B$=""
430 B=B/2:
440 IFB=INT(B)THENB$="0"+B$ELSEB$="1"+B$
450 B=INT(B):IFB<1THEN460ELSE430
460 PRINT"Binary"TAB(32-LEN(B$)):B$
470 IFC$="O"THENOS$=LEFT$(A$,E):GOTO530
480 O=D:O$=""
490 A=((O/8)-INT(O/8))*8
500 O=O-A
510 O$=CHR$(48+A)+O$
520 IFO>0THENO=O/8:GOTO490
530 PRINT"Octal"TAB(32-LEN(O$)):O$
540 IFC$="H"THENH$=LEFT$(A$,E):GOTO600
550 H=D:H$=""
560 A=((H/16)-INT(H/16))*16
570 H=H-A
580 IFA>9THENH$=CHR$(55+A)+H$ELSEH$=CHR$(48+A)+H$
590 IFH>0THENH=H/16:GOTO560
600 PRINT"Hexadecimal"TAB(32-LEN(H$)):H$
610 PRINT@287,"Press ENTER to do again."
;
620 IFINKEY$<>CHR$(13)THEN620ELSE20
630 FORM=ETO1STEP-1
640 E$=MID$(A$,M,1):
650 IFE$="0"THEN690
660 :IFE$>"/"ANDB$<:""THEND=D+VAL(E$)*C^(E-M):GOTO690
670 IFE$<"A"ORE$>"F"THENPRINT"Error in number":STOP
680 D=D+((ASC(E$)-64)+9)*C^(E-M)
690 NEXT:RETURN

```


MEMSCN

Dig around in computer memory to see what's there.

If you're interested in examining the information contained in your Model 100's ROM (Read Only Memory), or seeing your program and text files as they really appear in RAM (Random Access Memory), then you'll enjoy this program.

When you first run the program it'll ask you for the memory location you want to examine. Type in either a decimal or a hexadecimal number, using either a D or H to specify which it is.

The program will then display forty-eight consecutive memory locations, in six rows of eight columns. To help you keep your place, these rows are placed in the middle of the display. To the right of each line of eight bytes of information are the ASCII equivalents of each byte. To the left are the addresses of the first byte in each line. If you're confused, examine the following:

FFB8	00	0F	00	0F	00	37	00	35	7 5
FFC0	00	0D	00	0D	00	00	00	38	8
FFC8	00	38	00	38	00	38	00	38	8 8 8 8
FFD0	00	38	00	34	00	33	00	34	8 4 3 4

address
of
byte

information
in memory

ASCII
equivalent

At the bottom of the display are four letters: Q, D, H, X. These letters stand for Quit and return to beginning; change to Decimal display; change to Hexadecimal display; and convert high bit.

The Decimal option changes the hexadecimal numbers in the display to

decimal numbers. The hexadecimal number does the reverse. ASCII is unaffected by this change.

The last option is useful for examining ROM. Microsoft uses a special encoding technique to cut down on the amount of space that certain data lists require. In BASIC, a list of data in a DATA statement has to be separated by commas. This wastes one byte for each item in the list. In machine-language you can take advantage of the fact that the ASCII codes for the alphabet require only seven of the eight bits available in each location. Thus, as a programmer, you can use the eighth bit of each character as a sign to the program that this character marks the beginning of a new word of data, eliminating the need for a comma. This practice, of course, makes it difficult for you to read the list, since the first character of each data word is displayed as a non-alphabetic character. You could memorize the symbols that result from adding 128 to the ASCII codes (65-91), but that takes time. To simplify your work, the X key automatically sets the high bit of the bytes displayed to zero, letting you see the ASCII letter hidden by the high bit. Instead of just displaying the altered letters in normal video, I used reverse display (white character in dark background) so you can still tell where one word ends and another begins.

To see this at work, examine memory starting at location 80H (128D).

Program Listing for MEMSCN

```
5 REM 1449 bytes
10 CLS: CLEAR 256: PRINT TAB(10) "Memory Scan
   Program"
20 MAXFILES=0: PRINT: PRINT "1. Display Mem
   ory Block"
30 DEFSNGA-Z: PRINT "2. Return to Menu"
40 PRINT: INPUT "Your choice"; A$
50 A=VAL(A$): IFA=2 THEN MENU
60 IFA<>1 THEN 50
70 PRINT "Beginning Address (add H for he
   xa-"
80 Y=0: INPUT "decimal, D for decimal)"; A$
90 IFRIGHT$(A$,1)="D" OR RIGHT$(A$,1)="d" T
   HEN B=VAL(A$): Y=1: GOTO 120
100 IFRIGHT$(A$,1)<>"H" AND RIGHT$(A$,1)<>
   "h" THEN 70
110 GOSUB 390
120 IF B<0 THEN B=0 ELSE IF B>65480 THEN B=65480
130 CLS: FOR I=B TO B+48 STEP 8:
140 IF Y THEN PRINT USING "#### " ; I; ELSE GOSU
   B470: PRINT D$ " ";
```

```

150 FORJ=0TO7:C=PEEK(I+J):D$=""
160 IFXTHENIFCAND128THENI$=I$+CHR$(27)+"
    p"+CHR$(C-128)+CHR$(27)+"q":GOTO180
170 IFC>31THENI$=I$+CHR$(C)ELSEI$=I$+" "
180 GOSUB310:PRINTD$;
190 NEXT:PRINT" I$:I$="":NEXT
200 PRINT"Q, D, H, X, , ";
210 IFYTHENFORI=0TO54:FORJ=35TO180STEP18
    :PSET(J,I):NEXT:NEXT:FORI=0TO179:PS
    ET(I,55):NEXT
220 A$=INKEY$:IFA$=""THEN220
230 IFA$=CHR$(30)THENB=B+48:GOTO120
240 IFA$=CHR$(31)THENB=B-48:GOTO120
250 IFA$="x"ORA$="X"THENIFXTHENX=0:GOTO1
    20ELSEX=1:GOTO120
260 IFA$="q"ORA$="Q"THENRUN
270 IFA$="d"ORA$="D"THENY=1:GOTO120
280 IFA$="h"ORA$="H"THENY=0:GOTO120
290 GOTO220
300 REM convert decimal to hex
310 IFYTHEND$=RIGHT$(" "+STR$(C),3):RE
    TURN
320 D=CMOD16:E=C-D
330 IFD>9THEND$=CHR$(D+55)ELSE D$=CHR$(48
    +D)
340 IFE=0THEND$="0"+D$+" ":RETURN
350 E=E/16
360 IFE>9THEND$=CHR$(E+55)+D$:ELSE D$=CHR
    $(E+48)+D$
370 D$=D$+" ":RETURN
380 REM convert hex number to decimal
390 FORM=LEN(A$)-1TO1STEP-1
400 B$=MID$(A$,M,1):IFB$>"/"ANDB$<":"THE
    NB=B+VAL(B$)*(16^((LEN(A$)-1)-M)):G
    OTO440
410 IFB$<"A"OR(B$>"F"ANDB$<"a")ORB$>"f"
    THENPRINT"error in hex number":GOTO7
    0
420 IFB$>"F"THENB$=CHR$(ASC(B$)-32)
430 B=B+((ASC(B$)-64)+9)*(16^((LEN(A$)-1
    )-M))
440 NEXT:RETURN
450 REM convert decimal to hex
460 I=16383
470 IFI<256THENC=I:GOTO320
480 C=INT(I/256):GOSUB320
490 A$=D$:C=I-C*256:GOSUB320
500 D$=LEFT$(A$,2)+D$:RETURN

```


PRINT

Format your BASIC listing as you like.

This program takes an ASCII saved program (.DO format) and sends it to your lineprinter, letting you set the line lengths, page length, lines printed per page, starting page number, and the amount of indentation should a program line extend beyond the line length you specify.

If you wanted, you could add a feature to make the program break up multiple statement lines, indenting and putting each statement on a new line.

Program Listing for PRINT

```
10 REM 1176 bytes
20 MAXFILES=1: CLEAR513
30 PL=66: LP=54: LL=70: LI=5: P=1
40 CLS: PRINTTAB(13) "Program Formatter"
50 PRINT "1. Page Length = " PL
60 PRINT "2. Lines printed per page = " LP
70 PRINT "3. Line Length = " LL
80 PRINT "4. Line Indentation = " LI
90 PRINT "5. Starting page number = " P
100 INPUT "Do you want to change these"; A
    $
110 IFA$ = "" THEN 40
120 IFA$ = "N" OR A$ = "n" THEN 240
130 IFA$ <> "Y" AND A$ <> "y" THEN 40
140 PRINT @280, "Which one?";
150 A$ = INKEY$: IFA$ = "" THEN 150
160 A = VAL(A$): IFA < 1 OR A > 5 THEN 40
170 PRINT @280, "";
180 ON AGO SUB 190, 200, 210, 220, 230: GOTO 40
190 INPUT "Page length"; PL: RETURN
```



```

200 INPUT"Printed lines per page";LP:RET
    URN
210 INPUT"Line length";LL:RETURN
220 INPUT"Line indentation";LI:RETURN
230 INPUT"Starting page number";P:RETURN
240 CLS:PRINT:PRINT"Program must be in A
    SCII format (.do)"
250 LI$=STRING$(LI,32):TM=PL-LP
260 INPUT"Program name";P$
270 FORI=1TOLEN(P$):IFMID$(P$,I,1)>"Z"TH
    ENMID$(P$,I,1)=CHR$(ASC(MID$(P$,I,1
    ))-32)
280 NEXT
290 IFTM>3THENGOSUB500
300 OPENP$FORINPUTAS1
310 IFEOF(1)THENCLOSE:LPRINTCHR$(12):L=0
    :GOTO40
320 B=0:LINEINPUT#1,A$
330 A=LEN(A$)
340 L1$=LEFT$(A$,LL)
350 GOSUB440
360 IFA<=LLTHEN310
370 A$=RIGHT$(A$,A-LL)
380 A=LEN(A$)
390 L1$=LEFT$(A$,LL-LI)
400 B=1:GOSUB440
410 IFA<LL-LI THEN310
420 A$=RIGHT$(A$,A-(LL-LI))
430 GOTO380
440 IFBTHENLPRINTLI$;
450 LPRINTL1$:L=L+1:IFL>LPTHEN470
460 RETURN
470 L=1:IFTM=0THENRETURN
480 IFTM<4THENFORI=1TOTM:LPRINT:NEXT:RET
    URN
490 FORI=1TOTM/2:LPRINT:NEXT
500 LPRINTTAB((LL-(LEN(P$)))/2)P$;TAB(LL
    -8)USING"Page ###";P
510 FORI=1TO(TM/2-1):LPRINT:NEXT:P=P+1:R
    ETURN

```

PART VII

Sound

The Model 100 includes a sophisticated sound routine in BASIC that lets you specify 16383 different tones. There are many ways to incorporate this feature into programs. The most common is to use the sound function as a prompt to tell the user when something is waiting or an error was generated.

The programs in this section are different. They use the sound function as a teaching tool. The musical keyboard program, **MUSKEY**, lets you use the keyboard as a sound board. **MORSE**, the next program, teaches Morse code; and **MUSEC8** lets you create music.





MUSKEY

Make sounds by pressing keys.

This is a fun little program that changes all the keys to musical notes. The highest note is the "1" key, and the lowest is the "/" key, going from left to right on each row of the keyboard.

When you start, you can assign the length of each note an arbitrary length of up to 255. Using a number higher than this will result in an illegal function call error.

As an interesting sidelight, you can "play" messages. With the program running and waiting for your keyboard input, press any of the function buttons at the top of the keyboard. You'll hear each letter of the message assigned to the label "played" by the program. You can even use the PASTE button. From the MENU, go to text mode and type in a message. Use the Select and Arrow Key buttons to select a portion of the message, or all of it, for copying into the PASTE buffer. Now go back to the MENU and place the cursor over the MUSKEY program and press ENTER. BASIC will automatically load the program and start executing it. When the display indicates that the program is waiting for you to start playing keys, press the PASTE button. Your message will now be played for you. Using this technique, you can create and play songs.

Program Listing for MUSKEY

```
10 REM 416 bytes
20 DATA 1234567890-=
30 DATA QWERTYUIOP[
40 DATA ASDFGHJKL;'
50 DATA "ZXCVBNM,./"
60 DEFINT A-Z
70 DIM TV(91)
```

```

80 CLS:PRINTTAB(10)"Musical Keyboard"
90 INPUT"Tone length";TL
100 T=500
110 FORL=1TO4
120 READA$
130 FORX=1TOLEN(A$)
140 L$=MID$(A$,X,1)
150 TV(ASC(L$))=T
160 T=T+50
170 NEXT:NEXT
180 PRINT"Ok, play (press the ^ keyboard
    characterto start again).\"
190 A$=INKEY$:IFA$=""THEN190
200 IFA$="^"THENRUN
210 IFASC(A$)>91THENA$=CHR$(ASC(A$)-32)
220 SOUNDTV(ASC(A$)),TL
230 GOTO190

```

MORSE

Learn Morse Code.

If you've ever wanted to learn Morse Code, here's your chance. With this program you can either type characters on the keyboard and listen to the code assigned to each letter, or you can have the program randomly select a letter, sound the code, and wait for you to identify it. Either way it's an enjoyable and interesting way to learn.

You can control both the speed of the code transmission and the tone (low to high) used. Using the same technique described in the MUSKEY program, you can enter entire messages for play back.

If you have a friend who's also interested in the subject, you can take turns typing in messages for the other person to decipher. A fast and fun way to learn.

Program Listing for MORSE

```
5 REM 1941 bytes
10 CLS:PRINTTAB(10)"Morse Code Program":
   MAXFILES=0
20 CLEAR256:DATA A,.-
30 DATA B,-...
40 DATA C,-.-.
50 DATA D,-..
60 DATA E,.
70 DATA F,..-.
80 DATA G,--.
90 DATA H,....
100 DATA I,..
110 DATA J,----
120 DATA K,-.-
130 DATA L,-...
```

```

140 DATA M,--
150 DATA N,-.
160 DATA O,---
170 DATA P,---.
180 DATA Q,---.-
190 DATA R,-.-
200 DATA S,...
210 DATA T,-
220 DATA U,...-
230 DATA V,...-
240 DATA W,--
250 DATA X,-.-
260 DATA Y,-.-
270 DATA Z,---.
280 DATA 1,-----
290 DATA 2,...---
300 DATA 3,...--
310 DATA 4,....-
320 DATA 5,.....
330 DATA 6,-....
340 DATA 7,---...
350 DATA 8,----..
360 DATA 9,-----
370 DATA 0,-----
380 DATA .,-.-.-
390 DATA ":",---...
400 DATA /,-.-.-
410 DATA ",",---...
420 DATA 0,0
430 DIML,C$(90)
440 INPUT"TONE 1-3";TN:IFTN<1ORTN>3THEN4
    40
450 INPUT"SPEED 1-6";SP:IFSP<1ORSP>6THEN
    450
460 TN=TN*5460
470 READL$,C$
480 IFL$=""ANDC$=""THEN510
490 C$(ASC(L$))=C$
500 GOTO470
510 CLS:PRINT"Do you want to work with t
    he:"
520 PRINT:PRINT"1. MORSE CODE TYPER"
530 PRINT"2. MORSE CODE TESTER"
540 PRINT:PRINT"Your choice?"
550 A$=INKEY$:IFA$=""THEN550
560 IFA$="1"THEN590
570 IFA$="2"THEN710
580 GOTO550

```

```

590 CLS
600 PRINT"Press a letter (the only punc-
    "
610 PRINT"tuation supported is the colon
    "
620 PRINT "period, comma, and slash). P
    res"
630 PRINT CHR$(34)+"="+CHR$(34)+" to res
    tart the ";
640 PRINT"program."
650 A$=INKEY$:IFA$=""THEN650
660 IFA$=""THENRUN
670 V=ASC(A$):IFA$>="a"THENV=V-32
680 PRINTA$;" "C$(V)
690 GOSUB970
700 GOTO650
710 REM    TESTER
720 CLS:PRINTTAB(12);"Test starts now!"
730 PRINT"I will sound the code for a le
    tter,"
740 PRINT"You tell me which letter it is
    ."
750 PRINT"Press"+CHR$(34)+"="+CHR$(34)+"
    to restart,"
760 PRINT"Press SPACE BAR to hear code a
    gain."
770 PRINT:SD=VAL(RIGHT$(TIME$,2))
780 FORX=1TOSD
790 R=RND(1)
800 NEXT
810 V=INT(RND(1)*26)+64
820 IC=0:FORI=1TO500:NEXT
830 PRINT"What letter is this? ";
840 FORI=1TO500:NEXT:GOSUB970
850 A$=INKEY$:IFA$=""THEN850
860 IFA$="" THEN840
870 IFA$=""THEN RUN
880 VI=ASC(A$)
890 IFVI=VORVI=V+32THEN950
900 PRINT"Wrong."
910 IFIC<=3THENIC=IC+1:PRINT"Try again."
    :GOTO830
920 PRINT"Too many bad tries, that letter
    was: ";
930 PRINTCHR$(V)
940 PRINT:GOTO960
950 CLS:PRINT"Correct, very good."
960 PRINT"Next letter.":GOTO810

```


[illegible]

162

MUSIC8

Create your own tunes.

For the music aficionado, here's MUSIC8, a real music playing program.

It starts by drawing the five lines of a music sheet stanza and putting a graphic block in the upper left corner of the display. Use the arrow buttons to move the block up and down, and back and forth across the stanzas (you can't advance the arrow beyond the last note displayed). Holding down the arrow key will make it repeat and move rapidly up, down, left or right. When you have the graphic block at the proper position, you select the note for playing by typing Q for quarter notes, H for half notes, W for whole notes, and E for eighth notes. The "3" or "#" key is used for a sharp, and the "B" or "b" key is used for flats. As you select each note, the name of the note is displayed at the bottom of the screen.

When your musical composition is completed, press ENTER and the program will play it for you.

After playing, the program asks you if you want it to repeat your music. If you answer yes, it gives you the opportunity to change the tempo (speed) at which it plays. If you answer no, it re-displays the music and lets you alter it.

Unfortunately, there isn't enough room in memory for us to add a routine to let you save and load your compositions. Those skilled at programming may want to shorten this program and add the save and load routines themselves.

Program Listing for MUSIC8

```
10 REM 2420 bytes
20 CLEAR500
30 DIMA$(76)
40 Y=0
50 TA=20
60 CLS
```

```

70 GOSUB240
80 FORX1=0TO76
90 X=X1-INT(X1/39)*39
100 IFX=0ANDX1<>0THENQ$=INPUT$(1):IFQ$<>
    "#"ANDQ$<>"3"ANDQ$<>"B"ANDQ$<>"b"TH
    ENCLS:GOSUB240:Z1=X1+38:H=X:GOSUB10
    20:X=HELSEX=39:GOTO150
110 PRINT@INT(Y/2)*40+X,"";
120 Q$=INPUT$(1)
130 IFQ$=CHR$(13)THEN570ELSEIF(Q$=" "ORQ
    $=CHR$(28))ANDAS(X1)<>" "THENNEXTX1:
    GOTO570ELSEIF(Q$="W"ORQ$="H"ORQ$="Q
    "ORQ$="E")ANDAS(X1)<>" "THENGOSUB480
140 IFQ$="Q"THENGOSUB280:GOTO180ELSEIFQ$
    ="H"THENGOSUB380:GOTO180ELSEIFQ$="W
    "THENGOSUB410:GOTO180ELSEIFQ$="E"TH
    ENGOSUB490:GOTO180
150 IF(Q$="#"ORQ$="3")ANDX>0THENX=X-1:X1
    =X1-1:GOSUB530:Q$="#" :GOTO210ELSEIF
    (Q$="b"ORQ$="B")ANDX>0THENX=X-1:X1=
    X1-1:GOSUB540:Q$="b":GOTO210
160 IFQ$=CHR$(30)ANDY>0THENY=Y-1ELSEIFQ$
    =CHR$(31)ANDY<14THENY=Y+1ELSEIF(Q$=
    CHR$(8)ORQ$=CHR$(29))ANDX1>0THENX=X
    -1:X1=X1-1
170 IFX=-1THENCLS:GOSUB240:Z1=X1:GOSUB10
    20:X=38:GOTO110ELSE110
180 AS(X1)=RIGHT$(STR$(ABS(INT((Y+4)/7)-
    3)),1)+Q$+CHR$(71-Y-5+INT((Y+5)/7)*
    7)
190 NEXTX1
200 GOTO570
210 AS(X1)=AS(X1)+Q$
220 NEXTX1
230 GOTO570
240 FORA=0TO4
250 LINE(0,A*8+15)-(233,A*8+15)
260 NEXTA
270 RETURN
280 GOSUB550
290 FORA=0TO6
300 PSET(X*6+2,A+Y*4)
310 NEXTA
320 FORA=4TO6
330 FORB=0TO1
340 PSET(X*6+B,A+Y*4)
350 NEXTB
360 NEXTA

```

```

370 RETURN
380 GOSUB280
390 PRESET(X*6+1,5+Y*4)
400 RETURN
410 GOSUB550
420 LINE(X*6,Y*4)-(X*6+2,6+Y*4),1,B
430 PRESET(X*6,Y*4)
440 PRESET(X*6+2,Y*4)
450 PRESET(X*6,6+Y*4)
460 PRESET(X*6+2,6+Y*4)
470 RETURN
480 IFX=-1THENRETURNELSELINE(X*6,0)-(X*6
    +5,63),0,BF:AS(X1)="":GOTO240
490 GOSUB280
500 PSET(X*6+3,Y*4+1)
510 PSET(X*6+4,Y*4+2)
520 RETURN
530 IFX=-1THENRETURNELSEPSET(X*6+4,Y*4):
    PSET(X*6+4,Y*4+1):PSET(X*6+4,Y*4+2)
    :PSET(X*6+3,Y*4+1):PSET(X*6+5,Y*4+1)
    ):RETURN
540 IFX=-1THENRETURNELSEPSET(X*6+4,Y*4):
    PSET(X*6+4,Y*4+1):PSET(X*6+4,Y*4+2)
    :PSET(X*6+5,Y*4+1):PSET(X*6+5,Y*4+2)
    ):RETURN
550 PRINT@280+X,CHR$(71-Y-5+INT((Y+5)/7)
    *7);
560 RETURN
570 FORX=0TO76
580 AS=AS(X)
590 IFA$=""THEN680
600 O=2^(VAL(LEFT$(AS,1))-1)
610 N=(ASC(MID$(AS,3,1))-64)*2
620 IFRIGHT$(AS,1)="#"THENN=N+1
630 IFRIGHT$(AS,1)="b"THENN=N-1:IFMID$(A
    $,3,1)="G"THENO=2^(VAL(LEFT$(AS,1))
    -2)
640 L$=MID$(AS,2,1)
650 IFL$="W"THENL=TAELSEIFL$="H"THENL=TA
    /2ELSEIFL$="Q"THENL=TA/4ELSEL=TA/8
660 ONNGOTO720,740,760,780,800,820,840,8
    60,880,900,920,940,960,980,1000
670 SOUNDA/O,L
680 NEXTX
690 CLS
700 PRINT@120,"AGAIN (Y/N)?";
710 IFINPUT$(1)<>"Y"THENZ1=38:CLS:GOSUB2
    40:GOSUB1020:GOTO80ELSEPRINT@120,""

```

```

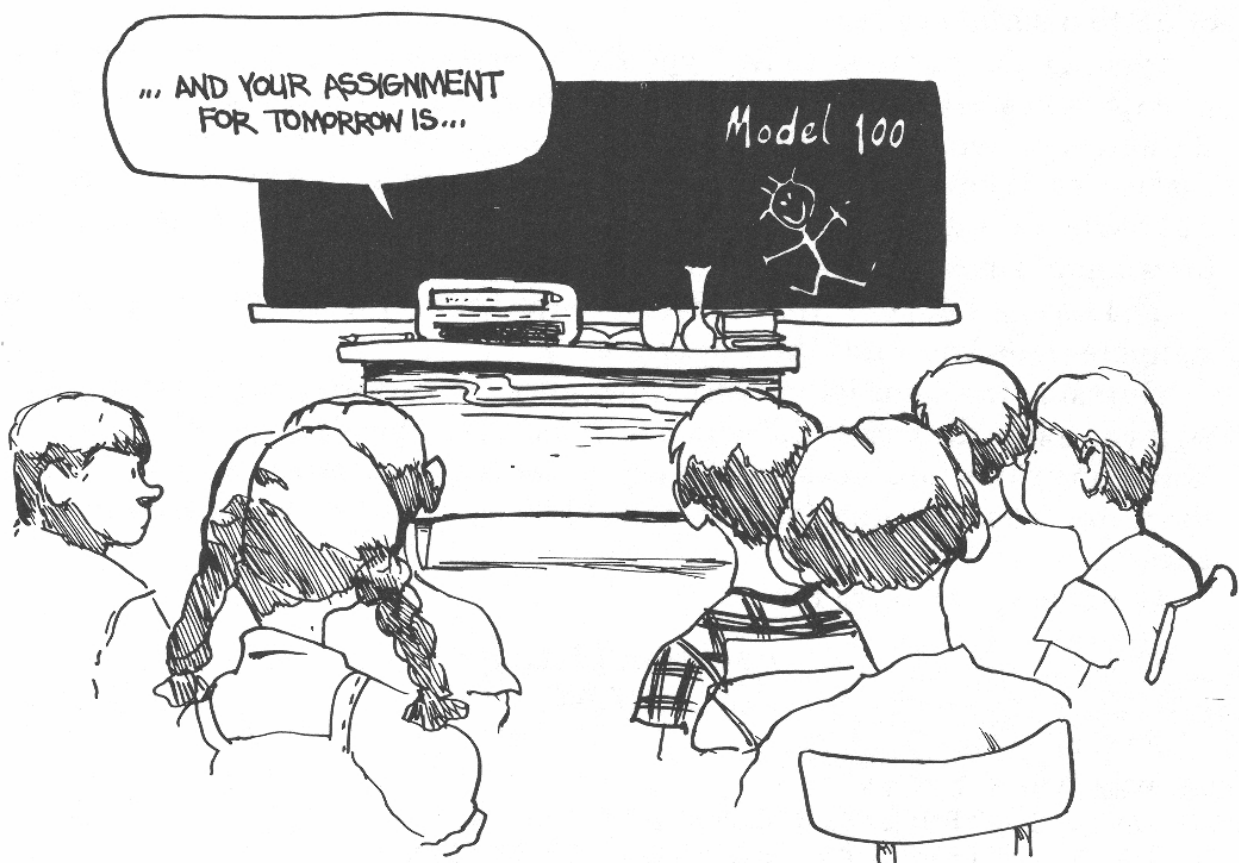
      ;:INPUT"TEMPO (4-100)";TA:IFTA<4ORT
      A>100THENTA=20:GOTO570ELSE570
720 A=11836
730 GOTO670
740 A=11172
750 GOTO670
760 A=10544
770 GOTO670
780 A=9952
790 GOTO670
800 A=0
810 GOTO670
820 A=9394
830 GOTO670
840 A=8866
850 GOTO670
860 A=8368
870 GOTO670
880 A=7900
890 GOTO670
900 A=7456
910 GOTO670
920 A=0
930 GOTO670
940 A=7032
950 GOTO670
960 A=6642
970 GOTO670
980 A=12538
990 GOTO670
1000 A=11836
1010 GOTO670
1020 FORQ=38TO0STEP-1
1030 A$=A$(Z1-Q)
1040 IFA$=""THEN1110
1050 X=38-Q
1060 Y=71-ASC(MID$(A$,3,1))+(3-VAL(LEFT$(
      (A$,1)))*7-5+INT(ASC(MID$(A$,3,1))/
      71)*7
1070 Q$=MID$(A$,2,1)
1080 IFQ$="W"THENGOSUB410ELSEIFQ$="H"THE
      NGOSUB380ELSEIFQ$="Q"THENGOSUB280EL
      SEGOSUB490
1090 Q$=RIGHT$(A$,1)
1100 IFQ$="#"THENGOSUB530ELSEIFQ$="b"THE
      NGOSUB540
1110 NEXTQ
1120 RETURN

```

PART VIII

Educational

Now we get to the younger, and not so young, set. These programs turn the Model 100 into a tutor for young children, teaching them spelling (HANG) and how to tell time (TIME). The next program in the section is a typing-teacher program (TYPE); both adults and children can benefit from this one. And the last program is also designed for all age groups, DRILL. It uses multiple choice, true/false, and fill-in-the-blank to teach word definitions.



HANG

Learn how to spell using Hangman.

Hangman is the children's perennial favorite game of spelling. To play the game, you first type in the words you want the child to learn in the DATA statements in lines 20-90. You can have up to 100 words in the program. More can be used by changing the A\$ variable in the dimension statement in line 110. If you're pressed for room, or don't need that many words, reduce the dimension of A\$ to a suitable size.

Playing the game is easy. Type RUN, press ENTER in response to the prompt, and start guessing the letters for the word. At the top of the display are the letters you've guessed already; underneath them is the word, with dashes for the blank unknown letters. Below that is the prompt asking for your guess. The right side of the display is used for drawing the hanging platform and your man (or woman, since this is a non-sexist game).

You have a total of fifteen wrong guesses before you lose the game, and the computer tells you what the word was.

Guessing the same letter twice is penalized only if the guess is wrong. That is, if you guess the letter "E," which is in the word, there's no penalty no matter how many times you press the "E" key. However, if you press "X," and it isn't in the word, then each time you press the "X" key it will be counted as one wrong guess.

Good luck, and have fun.

Program Listing for HANG

```
10 REM 1943 bytes
20 DATA CONTEMPTIBLE,CONSOLATION
30 DATA PACIFIC,OWNERSHIP,SOUTHWESTERN
```



```

40 DATA TRIAD,TRUCK,COMPUTER,ALARM CLOCK
50 DATA VISIBILITY,YANKEE,XYLOPHONE,YOUN
  GSTER
60 DATA RADIO SHACK,PARALLEL,PARENT
70 DATA INDELIBLE,COMMENDABLY,CHAIR
80 DATA SYSTEM,KEYBOARD,DICTIONARY
90 DATA REALISTIC,FEATURES,TABLE
100 DATA00
110 MAXFILES=0:CLEAR100:DIMA$(100),G(40)
120 GOSUB720:PRINT"Press ENTER to start.
  "
130 A$=INKEY$:R=RND(1):IFA$=""THEN130
140 N=1:PRINT@80,STRING$(30,32)
150 READA$(N)
160 IFA$(N)<>"00"THENN=N+1:GOTO150
170 N=N-1
180 REM START HERE
190 G$="":LG$=""
200 R=INT(RND(1)*N+1)
210 W$=A$(R)
220 FORX=1TOLEN(W$)
230 IFMID$(W$,X,1)=" "THENG$=G$+" ":GOTO
  250
240 G$=G$+"-"
250 NEXT
260 PRINT@40,"Letters you've guessed:":P
  RINTLG$
270 PRINT"The word:":PRINTG$
280 IFLEN(LG$)>15THEN530
290 PRINT"What's your next letter guess?
  ";
300 I$=INKEY$:IFI$=""THEN300
310 IFASC(I$)>90THENI$=CHR$(ASC(I$)-32)
320 T$=G$
330 C=0
340 G$=""
350 FORX=1TOLEN(W$)
360 IFMID$(W$,X,1)<>I$THENG$=G$+MID$(T$,
  X,1):GOTO380
370 G$=G$+I$:C=1
380 NEXT
390 IFC=0THENLG$=LG$+I$
400 ONLEN(LG$)GOSUB560,570,580,590,600,6
  10,620,630,650,660,670,680,690,700,
  710
410 C=0:FORX=1TOLEN(G$)
420 IFMID$(G$,X,1)="-"THENC=1
430 NEXT

```



```

440 IFC=1THEN260
450 CLS:PRINT:PRINTW$
460 PRINT"You win with: ";LEN(LG$);"wrong
    guesses."
470 PRINT"Do you want to play again (Y/N
    )?"
480 A$=INKEY$:IFA$=""THEN480
490 IFASC(A$)>90THENA$=CHR$(ASC(A$)-32)
500 IFA$="Y"THENGOSUB720:GOTO180
510 IFA$<>"N"THEN470
520 END
530 CLS:PRINT"You've made too many wrong
    guesses,"
540 PRINT"your word was: ";:PRINTW$
550 GOTO470
560 LINE(185,58)-(239,63),1,BF:RETURN
570 LINE(190,53)-(239,58),1,BF:RETURN
580 LINE(195,48)-(239,53),1,BF:RETURN
590 LINE(200,43)-(239,48),1,BF:RETURN
600 LINE(238,0)-(239,48),1,BF:RETURN
610 LINE(200,0)-(237,1),1,BF:RETURN
620 LINE(220,0)-(221,5),1,BF:RETURN
630 LINE(217,6)-(224,12),1,B
640 LINE(219,10)-(222,10):PSET(219,8):PS
    ET(222,8):RETURN
650 LINE(219,13)-(222,15),1,BF:LINE(215,
    16)-(226,30),1,BF:RETURN
660 LINE(216,31)-(218,40),1,BF:RETURN
670 LINE(225,31)-(223,40),1,BF:RETURN
680 LINE(226,39)-(228,40),1,BF:LINE(213,
    39)-(215,40),1,BF:RETURN
690 LINE(211,16)-(213,25),1,BF:LINE(211,
    26)-(212,28),1,BF:LINE(214,16)-(214
    ,18):RETURN
700 LINE(228,16)-(230,25),1,BF:LINE(229,
    26)-(230,28),1,BF:LINE(227,16)-(227
    ,18):RETURN
710 PSET(219,11):PSET(222,11):RETURN
720 CLS:PRINTTAB(16)"Hangman":PRINT:RETU
    RN

```

TIME

How to tell time with a big and small hand.

This program teaches a child how to read time. When you run it, it displays a standard clock face, with hour and minute hands. The short, fat hand is the hour hand, and the long thin one is the minute hand. You get five chances for getting the right answer; if you fail, the program tells you the time and goes to the next time display.

If you get tired of playing, you can press the "=" sign and end the program.

When you start the program you have the chance to set the accuracy with which the program will accept your answer. That is, you can play the game with a correct answer being anything within five minutes of the time selected by the program. Or two minutes, or one, whatever you want. This is especially helpful to children who can't accurately determine the number pointed to by the little hand of the clock (exact minutes aren't indicated).

Program Listing for TIME

```
10 REM 1358 bytes
20 MAXFILES=0:CLS:CLEAR256:D=3.14159/180
30 PRINT"Time Teacher":A$=STRING$(25,32)
40 INPUT"Maximum error for minutes";ME
50 PRINT@30,"12";STRING$(50,32):PRINT@31
   1,"6";
60 PRINT@185,"9";
70 PRINT@196,"3";
80 FORX=0TO360STEP9
90 S=SIN(X*D)
100 C=COS(X*D)
110 C=(C*20)+185
120 S=(S*20)+32
```

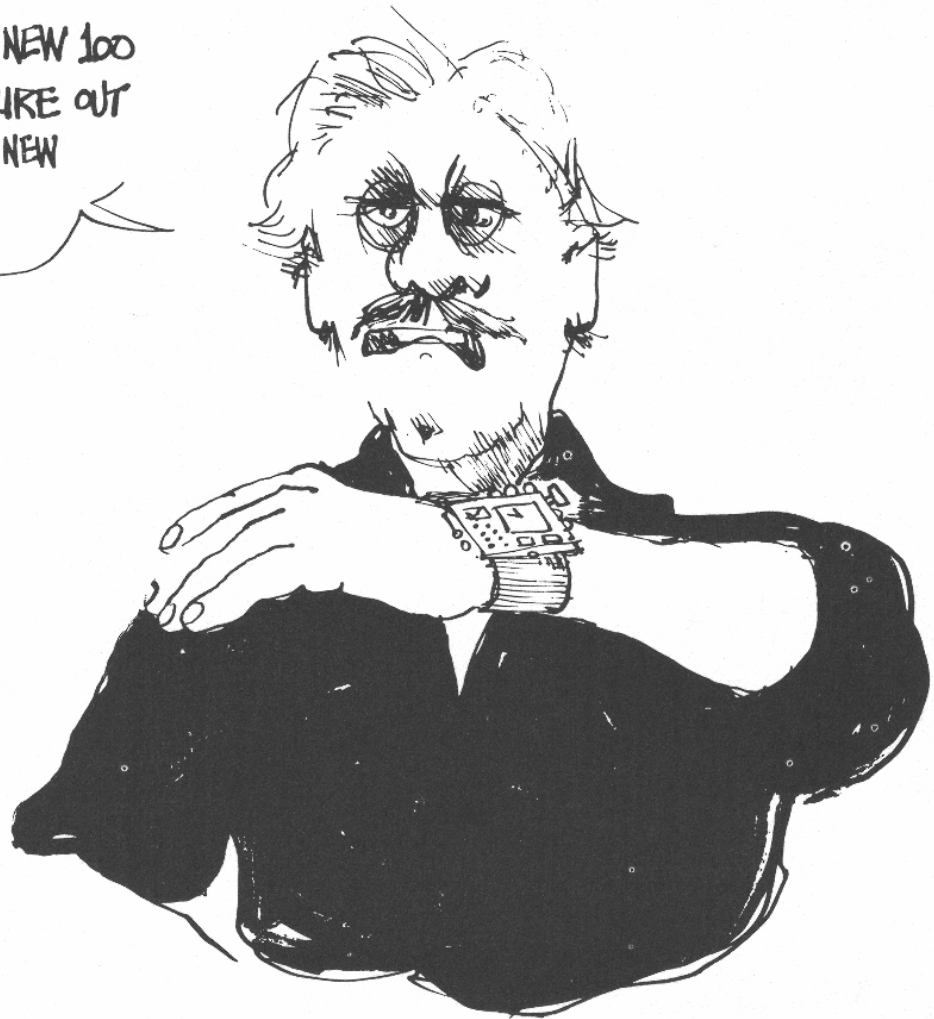
```

130 PSET(C,S)
140 NEXT
150 FORX=0TO330STEP30
160 SA=SIN(X*D):CA=COS(X*D)
170 S1=(SA*20)+32:S2=(SA*25)+32
180 C1=(CA*20)+185:C2=(CA*25)+185
190 LINE(C1,S1)-(C2,S2)
200 NEXT
210 PRINT@40,"Press any key to start"
220 PRINT"(Press '=' to stop.)"
230 B$=INKEY$:R=RND(1):IFB$=""THEN230
240 GOSUB590
250 XC=185:YC=32
260 M=INT(RND(1)*60)
270 H=INT(RND(1)*12)
280 MP=(M*100)/60
290 MA=((MP/100)*360)-90
300 C=(COS(MA*D)*18)+XC
310 S=(SIN(MA*D)*18)+YC
320 LINE(XC,YC)-(C,S)
330 MX=C:MY=S
340 HA=(H*30)-90
350 HA=HA+((MP/100)*20)
360 HY=(SIN(HA*D)*12)+YC
370 HX=(COS(HA*D)*12)+XC
380 LINE(XC,YC)-(HX,HY)
390 LINE(XC+2,YC)-(HX,HY)
400 LINE(XC,YC+2)-(HX,HY)
410 TK=0:PRINT@40,"What time is it?";
420 PRINT@80,"Hour";:INPUTHI$
430 IFHI$=""THENELSEHI=VAL(HI$)
440 INPUT"Minutes";MI
450 GOSUB600
460 IFH=0THENH=12
470 IFH<>HITHEN510
480 IFH=0THENH=12
490 ER=ABS(M-MI)
500 IFER<=METHEN610
510 PRINT@80,"WRONG"
520 FORLL=1TO1000:NEXT
530 TK=TK+1:IFTK<5THEN420
540 PRINT@120,"The time is:"H":M
550 PRINT"Press ENTER to continue."
560 IFINKEY$<>CHR$(13)THEN560
570 GOSUB600
580 GOTO630
590 PRINT@40,A$
600 PRINT@80,A$:PRINTA$:PRINTA$:PRINTA$:

```

```
      RETURN
610 PRINT@80,"That's right!"
620 FORL=1TO1000:NEXT
630 LINE(XC,YC)-(MX,MY),2
640 LINE(XC,YC)-(HX,HY),2
650 LINE(XC+2,YC)-(HX,HY),2
660 LINE(XC,YC+2)-(HX,HY),2
670 GOSUB600:GOTO260
```

I BOUGHT A NEW 100
TO HELP ME FIGURE OUT
HOW TO USE MY NEW
WATCH!



Ω

TYPE

Learn how to touch-type.

If you've always wanted to learn how to touch type, here's a program that will teach it to you.

The main difficulty with most touch typing books or courses is that you have to sit down in front of a typewriter, a noisy, paper wasting practice. You also have to schedule the time for this practice.

TYPE frees you from those constraints. Now you can practice, quietly, at your convenience. The program starts with the home row keys (asdfghjkl;) and goes through all four rows, including the shifted number keys. Each row, as you're practicing, has a graphic representation of the keys on the display. This will help break you of the habit of looking at the keys while typing, instead you can look at the display for hints on where to find the proper key.

In typing, your fingers should always rest on the home row. The left hand fingers are on the a, s, d and f keys (little finger on the a, index finger on the f), while the right hand fingers are on the j, k, l and ; keys (index finger on the j, little finger on the ;).

When you need to hit a key on a row above or below the home row, you move the appropriate finger that's on the home row up or down to hit the key. For example, use the left hand little finger for the z, a, q, 1 and ! keys. Use the right hand ring finger to hit the period, >, l, O, 9 and (keys.

When you need to shift, use the little finger of the opposite hand of the hand used to hit the shifted key. Use your thumbs to press the SPACE BAR.

Program Listing for TYPE

```
10 REM 3013 bytes
20 MAXFILES=0: CLEAR 256: CLS
30 PRINT TAB(10) "Typing Teacher"
```

```

40 PRINT"This program will familiarize y
   ou with"
50 PRINT"the M-100 keyboard layout. The
   keyboard";
60 PRINT"rows are counted from the botto
   m to the"
70 PRINT"top, thus the 'Z' key is in row
   one,"
80 PRINT"and the 'Q' key is in row three
   ."
90 PRINT" The second row, keys A to ',
   are"
100 GOSUB810:CLS
110 PRINT"called the Home Row Keys. Res
   t your
120 PRINT"fingers on these keys. Your l
   eft hand"
130 PRINT"Index finger should be on the
   'F' key.
140 PRINT"Your right hand index finger s
   hould be"
150 PRINT"on the 'J' key. Your left and
   right"
160 PRINT"little fingers will rest on th
   e A and L"
170 PRINT"keys, respectively."
180 GOSUB810:CLS
190 PRINT" Type each line exactly as it
   appears."
200 PRINT"Don't use the backspace key.
   Errors"
210 PRINT"will be indicated below your l
   ine"
220 PRINT"by an !"
230 PRINT"First, the Home Row."
240 GOSUB810:CLS
250 DATA"asdfghjkl;'", "zxcvbnm,./", "qwer
   tyuiop["
260 DATA"1234567890-=", " !@#$$%^&*()_+"
270 A=1
280 READA$:CLS:ONAGOSUB980,950,910,870,8
   70
290 PRINT"Your left index finger is used
   to strike the";
300 IFA=1THENPRINT"F and G";
310 IFA=2THENPRINT"V and B";
320 IFA=3THENPRINT"R and T";
330 IFA=4THENPRINT"4 and 5";

```

```

340 IFA=5THENPRINT"$ and %";
350 PRINT" keys, the right index finger
    is used on the";
360 IFA=1THENPRINT"H and J";
370 IFA=2THENPRINT"N and M";
380 IFA=3THENPRINT"Y and U";
390 IFA=4THENPRINT"6 and 7";
400 IFA=5THENPRINT"^ and &";
410 PRINT" keys."
420 IFA=1THEN470ELSEGOSUB810:GOSUB830
430 PRINT"The other fingers strike the k
    eys di- rectly ";
440 IFA=2THENPRINT" below ";ELSEPRINT" a
    bove ";
450 PRINT"the Home Row keys."
460 PRINT"The little fingers are used fo
    r shifting";
470 GOSUB810:GOSUB830
480 B$="":B=LEN(A$):
490 X=1:Y=B:Z=1:GOSUB840
500 X=B:Y=1:Z=-1:GOSUB840
510 X=1:Y=B:Z=1:GOSUB850
520 X=B:Y=1:Z=-1:GOSUB850
530 X=2:Y=B:Z=1:GOSUB860
540 X=B:Y=2:Z=-1:GOSUB860
550 C$=MID$(A$,RND(1)*B+1,1)
560 IFRND(1)<.18THENC$=" "
570 B$=B$+C$:IFLEN(B$)<35THEN550
580 GOSUB670
590 IFZ<1THENZ=Z+1:GOTO550
600 A=A+1:IFA=2THENPRINTTAB(15)"Row One"
610 IFA=3THENPRINTTAB(15)"Row Three"
620 IFA=4THENPRINTTAB(15)"Row Four"
630 IFA=5THENPRINTTAB(10)"Special Charac
    ters"
640 IFA<6THENGOSUB810:GOTO280
650 PRINT" You've successfully conclud
    ed Typing Teacher."
660 END
670 PRINT@160,B$:PRINT" ";
680 C$=INKEY$:IFC$=""THEN680
690 IFC$=CHR$(13)THENPRINTCHR$(8)" ":GOT
    O730
700 IFASC(C$)<32THEN680
710 L$=L$+C$
720 PRINTCHR$(8);C$;CHR$(95);:GOTO680
730 I=1:IFLEN(L$)=0THEN770
740 C=0:FORI=1TOLEN(L$)

```



```

750 IFMID$(B$,I,1)<>MID$(L$,I,1)THENPRIN
    TTAB(I-1)"!";:C=1
760 NEXT
770 IFLEN(L$)<LEN(B$)THENPRINTTAB(I-1)ST
    RING$(LEN(B$)-LEN(L$),"!");:C=1
780 PRINT:IFC=1THENFORI=1TO1000:NEXT
790 GOSUB830:L$="":B$="":RETURN
800 END
810 PRINT@287,"Press ENTER to continue."
    ;
820 IFINKEY$<>CHR$(13)THEN820ELSERETURN
830 PRINT@160,STRING$(153,32);:PRINT@120
    ,"":RETURN
840 FORI=XTOYSTEPZ:B$=B$+" "+MID$(A$,I,1
    ):NEXT:GOTO670
850 FORI=XTOYSTEPZ:C$=MID$(A$,I,1):B$=B$
    +" "+C$+C$:NEXT:GOTO670
860 FORI=X/2TOY/2STEPZ:C$=MID$(A$,I,2):B
    $=B$+" "+C$+MID$(A$,I+(Y/2),2):NEXT
    :GOTO670
870 PRINT@2,"1  2  3  4  5  6  7  8  9
    0  -  ="
880 IFA=5THENPRINT@2,"!  @  #  $  %  ^
    &  *  (  )  +"
890 LINE(5,7)-(220,7)
900 FORI=4TO220STEP18:LINE(I,0)-(I,7):NE
    XT
910 PRINT@42," Q  W  E  R  T  Y  U  I  O
    P  ["
920 LINE(11,7)-(209,7)
930 FORI=11TO215STEP18:LINE(I,8)-(I,15):
    NEXT
940 LINE(11,15)-(18,15)
950 PRINT@122,"   Z  X  C  V  B  N  M  ,
    .  /"
960 LINE(24,31)-(203,31)
970 FORI=23TO204STEP18:LINE(I,23)-(I,31)
    :NEXT
980 PRINT@82,"  A  S  D  F  G  H  J  K
    L  ;  '"
990 LINE(19,15)-(215,15):LINE(19,23)-(21
    5,23)
1000 FORI=18TO216STEP18:LINE(I,15)-(I,23
    ):NEXT
1010 PRINT@120,""
1020 RETURN

```

Ω

DRILL

Set up drill sessions on any subject.

I've called this program Foreign Language Drill because that's what I've used it for. You could use it for teaching children vocabulary words by changing the prompts asking for Foreign Word and Translation, to Word and Definition.

DRILL is a very useful program because you have several choices in how the words are given to you: Multiple choice; true/false; and fill-in-the-blank. You also control whether the word or its equivalent is given as the question, and how many questions you are given before you're rated on how you did.

When you start the program, it asks you for the words, and their equivalents. Type in the words until you have entered all that you want. Press ENTER at the first prompt to signify you have no more words to enter. The program will go to its menu and ask you if you want drill, multiple choice, true/false or fill-in-the-blank. After you select the method, you're asked how many questions you want. With these questions answered the program begins your drill.

The words are given to you in random order, without repetition, until either all the words have been given, or your question limit is reached.

To use this with children, you could either type in the words for the child, or replace the input routine with a READ...DATA statement routine (with the last method you wouldn't have to re-enter the words everytime you use the program).

Program Listing for DRILL

```
10 REM 2296 bytes free
20 MAXFILES=0: CLEAR 512: DIM A$(2,20)
30 GOSUB 760
40 PRINT "Type the foreign word, followed
    by its"
50 PRINT "translation. Enter Null for end"
```

```

      of list"
60 PRINT"(maximum of twenty word sets, f
   ive"
70 PRINT"minimum). "
80 I=1
90 PRINTI": ";B$="":LINEINPUT"Foreign Wo
   rd? ";B$
100 IFB$=""THEN140ELSEA$(1,I)=B$
110 C$="":LINEINPUT"English translation?
   ";C$
120 IFB$=C$ORC$=""THEN110ELSEA$(2,I)=C$
130 I=I+1:IFI<21THEN90
140 IFI<6THEN90ELSEI=I-1
150 REM delete the next 7 lines to save
   space.
160 GOSUB760:PRINT:PRINT"1. Multiple Cho
   ice"
170 PRINT"2. Fill in the Blank"
180 PRINT"3. True/False"
190 PRINT:PRINTTAB(13)"Your Choice";
200 INPUTB$:IFB$=""THEN160
210 B=VAL(B$):IFB<1ORB>4THEN160
220 CLS:INPUT"Which words do you want gi
   ven, English or Foreign";B$
230 IFB$=""THEN220
240 GOSUB770
250 IFB$="E"THENC=2:F=1ELSEC=1:F=2:IFB$<
   >"F"THEN220
260 INPUT"How many questions";B$:D=VAL(B
   $)
270 IFD=0ORD>ITHEN260
280 GOSUB720
290 ONBGOSUB360,560,620
300 PRINT"You made"A"mistake";:IFA<>1THE
   NPRINT"s";
310 PRINT", out of"D"questions."
320 PRINT"Do you want to use the same wo
   rds":INPUT"again";B$
330 A=0:IFB$=""THEN320ELSEGOSUB770
340 IFB$="N"THEN20ELSEIFB$<>"Y"THEN320
350 FORZ=1TOI:A$(0,Z)="":NEXT:GOTO160
360 FORY=1TOD:GOSUB790
370 PRINT:FORH=1TO4
380 G=INT(RND(1)*I)+1
390 IFA$(0,G)="1"ORA$(0,G)="3"THEN380
400 IFA$(0,G)=""THENA$(0,G)="1"
410 IFA$(0,G)="2"THENA$(0,G)="3"
420 K(H)=G

```

```

430 NEXT
440 H=RND(1)*4+1:IFA$(0,K(H))="1"THENAS(
    0,K(H))=""
450 IFA$(0,K(H))="3"THENAS(0,K(H))="2"
460 K(H)=E
470 FORH=1TO4:PRINTTAB(10)H;A$(F,K(H)):N
    EXT
480 INPUT"Your Choice";B$:J=VAL(B$)
490 IFJ<1ORJ>4THENGOSUB780:GOTO480
500 IFK(J)<>ETHENX=1:PRINT@280,"Wrong.":
    FORH=1TO500:NEXT:GOSUB780:GOTO480
510 IFX=1THENA=A+1
520 FORH=1TO4:IFK(H)=ETHENAS(0,K(H))="2"
    :GOTO550
530 IFA$(0,K(H))="3"THENAS(0,K(H))="2":G
    OTO550
540 A$(0,K(H))=""
550 NEXT:GOSUB720:NEXT:RETURN
560 FORY=1TOD
570 GOSUB790
580 PRINT@120,"":LINEINPUT"What? ";B$:IF
    B$=""THEN580
590 IFB$=A$(F,E)THENPRINT"Right.":FORH=1
    TO50:NEXT:GOTO610
600 PRINT"Wrong.":FORH=1TO500:NEXT:A=A+1
    :PRINT@120,STRING$(120,32):GOTO580
610 GOSUB720:NEXT:RETURN
620 FORY=1TOD
630 GOSUB790:A$(0,E)="2"
640 IFRND(1)>.5THENG=E:GOTO660
650 G=INT(RND(1)*I)+1:IFA$(0,G)="1"THEN6
    50
660 PRINTA$(F,G)". "
670 PRINT@120,"";:INPUT"True or False";B
    $:IFB$=""THEN670
680 GOSUB770
690 IF(B$="F"ANDG<>E)OR(B$="T"ANDG=E)THE
    NPRINT"Right.":FORH=1TO50:NEXT:GOTO
    710
700 PRINT"Wrong.":FORH=1TO500:NEXT:A=A+1
    :PRINT@120,STRING$(120,32):GOTO670
710 A$(0,E)="1":GOSUB720:NEXT:RETURN
720 CLS:ONBGOTO730,740,750
730 PRINTTAB(12)"Multiple Choice":RETURN
740 PRINTTAB(11)"Fill in the Blank":RETU
    RN
750 PRINTTAB(13)"True or False":RETURN
760 CLS:PRINTTAB(9)"Foreign Language Dri

```

```

11":RETURN
770 B$=LEFT$(B$,1):IFB$>"Z"THENB$=CHR$(A
    SC(B$)-32):RETURNELSERETURN
780 PRINT@240,STRING$(20,32):PRINT@280,S
    TRING$(20,32);:PRINT@240,"";:RETURN

790 E=INT(RND(1)*I)+1
800 IFA$(0,E)="1"THEN790
810 PRINTCHR$(34)A$(C,E)CHR$(34)" means:
    ";:A$(0,E)="1"
820 RETURN

```

SO... YA SPEAK
ANY FOREIGN
LANGUAGES
OR WHAT?



The Model 100 Program Book

**By
Terry Kepner and David Huntress**

- A collection of 51 short, interesting and useful BASIC programs for home, office and educational uses.
- These programs were specifically designed to take advantage of the Model 100 portable computer.
- The programs illustrate programming methods and techniques for beginning users of this fascinating computer.

WSi WEBER
SYSTEMS
INCORPORATED